



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FINAL DE GRADO

---

# **Implementación de un módulo de envío de emails personalizado para un proyecto de Inbound Marketing**

---

*Realizado por:*  
Laura Marín Serrano

*Supervisado por:*  
Lledó Museros Cabedo

*Tutorizado por:*  
Carlos Sánchez Castillo

Fecha de lectura: 13/10/2020  
Curso académico: 2019/2020

## **Resumen**

En este proyecto se expondrá el diseño, planificación e implementación del trabajo de final de grado para el grado de Ingeniería Informática llevado a cabo por la alumna Laura Marín Serrano durante su estancia en prácticas en la empresa *Aticsoft*, localizada en Castellón de la Plana.

El proyecto que se va a llevar a cabo es el desarrollo de un módulo de automatización de emails. Con esto se pretende que la forma de redactar emails sea mucho más rápida que la tradicional, ya que se le proporcionarán una serie de módulos al cliente que le brindarán la posibilidad de ir modificándolos a su gusto sin necesidad de invertir más tiempo en la estructura de estos.

## **Abstract**

In this project, the design, planning and implementation of the final degree project for the Computer Engineering degree carried out by the student Laura Marín Serrano will be exposed during her internship at the company *Aticsoft*, located in Castellón de la Plana.

The project to be carried out is the development of an email automation module. With this, it is intended that the way of writing emails is much faster than the traditional one, since a series of modules will be provided to the client that will give them the possibility of modifying them to their liking without having to invest more time in the structure of these.

## **Palabras clave**

Ajax, plantillas, email.

## **Keywords**

*Ajax, templates, email.*

# Índice general

<b>Introducción</b>	<b>9</b>
1.1. Contexto y motivación del proyecto	9
1.1.1. Descripción de la empresa	9
1.1.2. Motivación del proyecto	9
1.2. Objetivos del proyecto	10
1.3. Estructura de la memoria	11
<b>Descripción y planificación del proyecto</b>	<b>13</b>
2.1. Metodología	13
2.2. Objetivos y alcance	14
2.3. Tecnologías y herramientas	15
<b>2.4. Análisis de riesgos</b>	<b>21</b>
2.5. Planificación	21
2.5.1. Planificación inicial	22
2.5.2. Planificación final	23
2.6. Estimación de recursos y costes del proyecto	26
2.7. Seguimiento del proyecto	27
2.7.1. Sprint 1	28
2.7.1.1. Planificación del Sprint 1	28
2.7.1.2. . Planificación del Sprint 2	28
2.7.2. Sprint 2	28
2.7.2.1. Planificación del Sprint 2	28
2.7.2.2. Planificación del Sprint 3	29
2.7.3. Sprint 3	29
2.7.3.1. Planificación del Sprint 3	29
2.7.3.2. Planificación del Sprint 4	29
2.7.4. Sprint 4	29
2.7.4.1. Planificación del Sprint 4	29
2.7.4.2. Planificación del Sprint 5	29
2.7.5. Sprint 5	30
2.7.5.1. Planificación del Sprint 5	30
2.7.5.2. Planificación del Sprint 6	30
2.7.6. Sprint 6	30
2.7.6.1. Planificación del Sprint 6	30
<b>Análisis y diseño del sistema</b>	<b>30</b>
3.1. Definición de requisitos	31
3.2. Análisis del sistema	34
3.2.1. Diagrama de casos de uso	34
3.2.2. Diagrama de actividades	39
<b>3.3. Diseño de la arquitectura</b>	<b>40</b>

3.4. Diseño de la interfaz	41
3.5. Diseño de la base de datos	42
<b>Implementación</b>	<b>44</b>
4.1. Detalles de la implementación	44
4.2. Estructura de la aplicación	45
4.3. Resultados de la aplicación	47
4.4 Dificultades de implementación	63
<b>Pruebas</b>	<b>64</b>
<b>Conclusiones</b>	<b>66</b>
<b>Anexo</b>	<b>68</b>
<b>Bibliografía</b>	<b>70</b>

# Índice de figuras

[Figura 1: Cálculo peor caso.](#)

[Figura 2: Cálculo peor caso.](#)

[Figura 3: Diagrama de casos de uso.](#)

[Figura 4: Diagrama de actividades.](#)

[Figura 5: Arquitectura 3 capas.](#)

[Figura 6: Esquema de navegación del cliente.](#)

[Figura 7: Contenido tabla templates.](#)

[Figura 8: Estructura del proyecto.](#)

[Figura 9: Pantalla inicio.](#)

[Figura 10: Menú inicio.](#)

[Figura 11: Submenú diseño correo expandido.](#)

[Figura 12: Diseño correo: Todas las plantillas.](#)

[Figura 13: Diseño correo: Cabeceras.](#)

[Figura 14: Diseño correo: Cuerpos.](#)

[Figura 15: Diseño correo: Pies.](#)

[Figura 16: Vista plantilla usando 'drag and drop'.](#)

[Figura 17: Estilos de correo sin selección de plantilla.](#)

[Figura 18: Opciones estilo cabecera.](#)

[Figura 19: Opciones estilo cuerpo.](#)

[Figura 20: Opciones estilo pie de página.](#)

[Figura 21: Estilos correo con selector de color visible.](#)

[Figura 22: Plantillas originales.](#)

[Figura 23: Plantillas editadas.](#)

[Figura 24: Vista previa.](#)

[Figura 25: Vista previa pantalla completa.](#)

[Figura 26: Vista previa móvil.](#)

[Figura 27: Modal envío correo.](#)

[Figura 28: Proceso correcto de envío.](#)

[Figura 29: Correo recibido.](#)

[Figura 30: Prototipo de vista de módulos.](#)

[Figura 31: Prototipo de vista de estilos.](#)

[Figura 32: Prototipo de visualización de pantalla móvil.](#)

[Figura 33: Prototipo de envío de correo.](#)

# Índice de tablas

[Tabla 1: Planificación inicial del proyecto.](#)

[Tabla 2: Planificación final del proyecto.](#)

[Tabla 3: Tabla resumen del coste total del proyecto.](#)

[Tabla 4 : Requisito 01.](#)

[Tabla 5 : Requisito 02.](#)

[Tabla 6: Requisito 03.](#)

[Tabla 7: Requisito 04.](#)

[Tabla 8 : Requisito 05.](#)

[Tabla 9 : Requisito 06.](#)

[Tabla 10 : CU01: Consultar plantillas.](#)

[Tabla 11 : CU02: Consultar plantillas.](#)

[Tabla 12 : CU03: Consultar plantillas.](#)

[Tabla 13: CU04: Consultar plantillas.](#)

[Tabla 14: CU05: Consultar plantillas.](#)

[Tabla 15: Funciones administrador.](#)





# Capítulo 1

## Introducción

### 1.1. Contexto y motivación del proyecto

En esta sección se describe la empresa y el proyecto propuesto, así como la necesidad y motivación por la que se ha llevado a cabo.

#### 1.1.1. Descripción de la empresa

Este proyecto se desarrolló en una empresa llamada Aticsoft, localizada en Castellón de la Plana. Aticsoft se encuentra instalada en el Espaitec. Se trata de un parque Científico, Tecnológico y Empresarial de la Universidad Jaume I de Castelló formado por dos edificios. Es un espacio innovador compuesto por empresas de base tecnológica, grupos de investigación, asociaciones y entidades públicas.

Aticsoft se encuentra en el edificio Espaitec 2 y la empresa se dedica básicamente al desarrollo web, más concretamente, se trata de una compañía de desarrollo cloud con experiencia en sistemas orientados al marketing online, social media y ERP's verticales. Está especializada en los sectores de cerámica, publicidad y retail, utilizando como lenguajes base a la hora de programar SQL y PHP.

Se trata de una empresa pequeña formada por 8 empleados, de los cuales, 6 son desarrolladores, 1 es el CEO, y 1 se encarga de la parte de red.

La empresa surgió hace 15 años, en 2005, y ya cuenta con más de 1200 proyectos realizados con éxito. Algunos de los proyectos que ha realizado han sido por ejemplo; desarrollo de la página web y facebook para el festival del Arenal Sound que tiene lugar en la playa de Burriana en verano, desarrollo web para el festival de les Arts en Valencia y ha formado parte en el rediseño de la web de la cámara de Valencia.

### 1.1.2. Motivación del proyecto

La motivación de este proyecto de cara a la empresa surgió de la necesidad de conseguir un software capaz de automatizar o facilitar el envío de correos publicitarios de manera masiva.

Tras el parón producido a causa de la pandemia del virus Covid-19 el proyecto cambió de golpe. La nueva perspectiva era de un software que facilitara a los distintos usuarios el envío de emails personalizables. Gracias a este software, el usuario será capaz de diseñar el cuerpo del mensaje con estilos complejos a través de las distintas plantillas disponibles. Una vez aplicadas las plantillas deseadas, el usuario tendrá que ser capaz de personalizar los colores, textos, enlaces e imágenes de estas a su gusto de una manera fácil e intuitiva.

Esta sencillez a la hora de crear contenidos visuales complejos, permitirá a la empresa obtener un aumento de clientes gracias a la simplicidad ofrecida con dicho servicio.

## 1.2. Objetivos del proyecto

El objetivo principal de este proyecto es desarrollar un módulo de automatización de emails. De esta forma el envío de mensajes será mucho más rápido ya que no se tendrá que pensar en la estructura del mensaje.

Se proporcionarán una serie de plantillas diferentes al usuario; cabeceras, cuerpos y pies de páginas, las cuales vendrán con una estructura y texto por defecto.

El cliente podrá seleccionar los que más le convengan y modificarlos a su gusto, escribiendo el texto que crea oportuno o cambiando las imágenes por archivos de su ordenador.

Los objetivos que tendrían que ver con la parte del administrador, se podrían clasificar en los siguientes puntos:

- Generar una lista de plantillas en base de datos para cargarlas dinámicamente.
- Generar la estructura de la web con un panel de herramientas lateral.
- Programar las funciones de obtención de datos así como de envío de correos.
- Permitir al usuario la interacción total con la web en todo lo que la edición del correo conlleva.
- Sincronizar distintos lenguajes como html, css, sql, javaScript y PHP.

Los objetivos que tendrían que ver con la parte del cliente/usuario, se podrían clasificar en los siguientes puntos:

- Añadir plantillas mediante el uso de tecnologías como "Drag and Drop" o mediante clicks en ellas.

- Modificar el texto de estas y cambiar las imágenes que contengan.
- Editar tanto los colores de fondo como los de la tipografía.
- Visualizar previo envío el contenido del correo en formato desktop o mobile.
- Enviar un correo.

Con los objetivos mencionados anteriormente, se consiguen una serie de beneficios:

- Comodidad por parte de los clientes a la hora de redactar los mensajes, esto hará que prefieran usar nuestra aplicación de cara a otras aplicaciones de envío de mensajes tradicionales.
- Interfaz apropiada al usuario ya que es muy intuitiva.

## 1.3. Estructura de la memoria

Esta memoria recoge la información acerca del desarrollo del proyecto formativo durante la estancia en prácticas.

Para una mejor organización del contenido de este documento, voy a explicar qué información va a contener cada capítulo.

Para empezar, en el capítulo 1, se ha descrito brevemente a la empresa donde he realizado las prácticas y la motivación que ha llevado a realizar el proyecto. Por último, se han explicado los objetivos que tiene el proyecto.

En segundo lugar, en el capítulo 2, se explica mucho más detalladamente el proyecto, así como las tecnologías y herramientas usadas.

En tercer lugar, en el capítulo 3, se explica la planificación que ha tenido lugar para poder realizar el proyecto con éxito y dentro de los plazos establecidos. Se muestra información acerca de la metodología usada, así como información acerca de los recursos y costes estimados y del seguimiento del proyecto.

En cuarto lugar, en el capítulo 4, se detalla el análisis y diseño del sistema, así como el de la base de datos.

En quinto lugar, en el capítulo 5, se habla de la implementación, es decir, se describe el código implementado, además de comentar los resultados y dificultades durante el desarrollo del mismo.

En sexto lugar, en el capítulo 6, se exponen las conclusiones obtenidas tras el proceso de desarrollo.

Finalmente se expone la bibliografía que ha sido utilizada a la hora de realizar el proyecto.

## Capítulo 2

# Descripción y planificación del proyecto

En este segundo capítulo expondré la planificación que he seguido en el proyecto. Hablaré de la metodología usada en el proyecto así como también del alcance y objetivos de este. Luego explicaré las tecnologías y herramientas que he usado y haré una estimación de los costes y recursos antes de finalizar con un seguimiento del mismo.

### 2.1. Metodología

Para llevar a cabo el proyecto, se utilizó la metodología de desarrollo ágil SCRUM [1]. Esta metodología se basa en los sprints, los cuales suelen tener una duración de 15 días entre uno y otro. El sprint es el periodo en el cual se desarrolla el trabajo, al final de cada sprint se deben presentar los avances logrados, además, el cliente puede ir viendo los progresos y su *feedback* puede servir para realizar posibles mejoras.

Inicialmente, las reuniones tenían lugar en la empresa. Tras el comienzo de la cuarentena, provocada por la situación mundial en la que nos encontrábamos por la pandemia del COVID-19, las reuniones se realizaron a través de Skype.

Scrum propone tres roles principales, los cuales son:

- *Product Owner*: se asegura de que el equipo trabaje correctamente desde la perspectiva de negocio.
- *Scrum Master*: se encarga de eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint.
- *Scrum Team*: es el equipo de desarrollo el cual se encarga de realizar la implementación del proyecto.

Los dos primeros roles los llevó a cabo el supervisor del proyecto. El *Scrum Team* estará formado por el alumno, el cual es el encargado de desarrollar el proyecto.

A parte de los tres roles explicados anteriormente, tenemos los roles auxiliares, los *stakeholders*. Estos no se involucran frecuentemente en el proceso, pero deben ser tenidos en cuenta ya que son las personas que hacen posible el proyecto y para quienes el proyecto producirá el beneficio acordado. Dentro de esta categoría, se encuentran personas tales como los clientes, proveedores, vendedores...

En mi proyecto estos roles los ha asumido Carlos (mi supervisor directo).

Paralelamente, se realizaba un Daily Standup, también llamado Scrum diario. Se trata de pequeñas reuniones que tenían lugar diariamente entre los miembros del equipo, en mi caso entre el jefe que se encargaba de desempeñar los dos roles, el de *Product Owner* y *Scrum Master* y yo que tenía el rol de *Scrum Team*. Estas reuniones no duraban más de 5 minutos, se realizaban todas las mañanas a las 9 menos 5, es decir, 5 minutos antes de que empezase a desarrollar el proyecto. El objetivo de estas reuniones es mantenerse actualizado sobre el trabajo realizado desde el último standup, hablar de los problemas encontrados o que se piensan que podrían encontrarse, y proponer soluciones.

## 2.2. Objetivos y alcance

Inicialmente, mi proyecto iba a consistir en un módulo de automatización de emails, pero tras las circunstancias en las que nos encontramos causadas por la pandemia del virus Covid-19 que generaron una interrupción en el desarrollo de las prácticas para posteriormente finalizarlas teletrabajando, nos vimos en la necesidad de tener que modificar la idea inicial. Con este cambio el proyecto se basaría en un software de envío de emails creados a partir de plantillas tanto personalizadas como personalizables.

Está destinado a cualquier usuario que quiera enviar emails de una forma rápida e intuitiva.

La página web consta de una pantalla de inicio, donde se gestiona toda la funcionalidad desde la misma. Cuando el usuario quiere redactar un mensaje, puede visualizar todas las plantillas disponibles y únicamente tiene que seleccionar el módulo que quiera, ya sea seleccionándolo o arrastrándolo al espacio dispuesto para este (Drag and Drop).

A parte, como las plantillas están predefinidas, puede cambiar el texto, los colores y las imágenes a su gusto para redactar el mensaje como desee. De esta forma, proporcionamos un control completo al usuario con el que podrá decidir en todo momento si tiene que prescindir de alguna plantilla ya escogida o si tiene que añadir plantillas extra a las distintas secciones.

Si el usuario desea dejar de redactar el mensaje para empezar de nuevo o elegir otras plantillas diferentes, también tiene una opción disponible que le permitirá el borrado completo de las plantillas que había seleccionado previamente.

Una vez tenga el mensaje terminado, si presionamos la opción de enviar correo, se mostrará un modal donde podremos introducir tanto el asunto del correo como el destinatario al cual va dirigido el mensaje. Por debajo, se hace una gestión de las excepciones, es decir, se controlan los posibles fallos que se puedan producir para que el usuario siempre tenga la posibilidad de volver a atrás.

A su vez, siempre tendremos la opción de poder visualizar tanto en formato desktop como en formato móvil el contenido seleccionado hasta el momento.

Por último, encontraremos un poco de información profesional acerca del desarrollador.

## 2.3. Tecnologías y herramientas

En este apartado, voy a hablar de todas las tecnologías y herramientas de las cuales he hecho uso a lo largo de este proyecto. Explicaré en qué consisten, y el papel que han tomado dentro del proyecto.

Algunas de las herramientas empleadas para realizar el proyecto han sido definidas desde el inicio por la empresa, mientras que otras las he ido utilizando yo por mi cuenta al ver que eran necesarias para poder llevarlo a cabo de manera satisfactoria.

Voy a empezar hablando de los lenguajes usados a la hora de desarrollar la aplicación.

### Php

*PHP* (acrónimo recursivo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular ejecutado en el lado del servidor, especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. [\[1\]](#)

He utilizado este lenguaje en el lado del servidor (backend). Para el correcto flujo de la aplicación he optado por generar dos archivos; el primero de ellos (ajaxphp.php) tiene la importancia de actuar como puente de comunicación entre el servidor y el cliente. Contactando a través de ajax con este archivo obtenemos los datos necesarios. En el segundo archivo (funcionesbbdd.php) he generado toda función relacionada con la base de datos así como la función de envío de emails.

### SQL

SQL (Structured Query Language) es un lenguaje de dominio específico utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales. [\[2\]](#)

En el proyecto, la base de datos utilizada ha sido creada usando el lenguaje SQL, las plantillas que he generado para el envío de emails, se han introducido en la base de datos de MySQL a través de PHPMyAdmin.

## Ajax

AJAX, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. Gracias a esto, es posible realizar cambios sobre las páginas sin necesidad de recargarlas, produciendo una mejora de la interactividad, velocidad y usabilidad en las aplicaciones.

Se trata de una tecnología asíncrona, es decir, los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. [\[3\]](#)

Por medio de Ajax, conseguimos obtener o enviar de manera dinámica los siguientes datos:

- Listado o catálogo de plantillas disponibles para el usuario.
- Obtención del contenido de estas plantillas una vez se han seleccionado.
- El envío del contenido del correo, el asunto y el destinatario al servidor para su posterior envío.

## HTML

HTML, siglas en inglés para HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un lenguaje muy básico y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos.

Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web.

HTML se considera el lenguaje web más importante, se trata de el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores han adoptado. [\[4\]](#).

En el proyecto, HTML ha sido usado para definir la estructura de la parte visual de nuestro sitio web.

## CSS

CSS (siglas en inglés de Cascading Stylesheets) es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.



Es muy usado para definir el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML.

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o *layouts*, los colores y las fuentes. [\[5\]](#).

En el proyecto, he usado CSS para establecer el diseño del HTML. Gracias al CSS ha podido tener un aspecto visual bastante agradable.

A continuación se presentan los frameworks utilizados.

## **Bootstrap**

Bootstrap es un framework web para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. Se ocupa únicamente del desarrollo front-end [\[6\]](#).

Gracias a Bootstrap hemos conseguido ahorrar tiempo y código a la hora de aplicar estilos a los diferentes elementos html por medio de las clases que nos proporciona.

## **JQuery**

Es una biblioteca de JavaScript. que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web [\[7\]](#).

Gracias a JQuery todas las funciones de JavaScript son más claras y simples.

## **JQuery-ui**

JQuery UI es una biblioteca de componentes para el framework jQuery que le añaden un conjunto de plug-ins, widgets y efectos visuales para la creación de aplicaciones web [\[8\]](#).

Utilizando JQuery-ui hemos conseguido manejar distintos objetos con los eventos drag and drop así como otras animaciones.

## Color-picker Farbtastic

Es un plugin JQuery que permite añadir widgets de selectores de color en la página a través de JavaScript [\[9\]](#).

Su implementación se basa en el uso del selector de color a la hora de modificar una plantilla tanto en los colores del texto como en los fondos de pantalla.

## Eclipse

Eclipse es una plataforma de software compuesta por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar código [\[10\]](#).

En concreto, he tenido que descargar los paquetes de IDE para desarrolladores de PHP, incluyendo editores para JavaScript, HTML, CSS y XML.

Para poder instalar el eclipse correctamente, previamente me he tenido que descargar e instalar Java Runtime Environment

Una vez instalado el desarrollo de la página se ha realizado completamente a través de él.

## Chrome DevTools

El Chrome DevTools (Inspector de elementos) es una herramienta de desarrollo integrada en el buscador de Chrome. Es utilizada para diagnosticar posibles problemas o errores. También permite testear las modificaciones mínimas para ser visualizadas en pantalla en tiempo real. Tiene diferentes paneles, cada uno de los cuales tiene una funcionalidad diferente [\[11\]](#).

A la hora de realizar mi proyecto, he estado prácticamente utilizando esta herramienta en todo momento gracias a su gran utilidad.

Con él he podido realizar las siguientes acciones:

1. Modificar en tiempo real tanto el código html, CSS, así como JavaScript. De esta manera conseguir unos estilos CSS es más rápido e intuitivo.
2. El uso de la consola como medio de depuración. En ella comprobaba si los selectores eran correctos y alguna que otra función de javaScript cuyo funcionamiento no estaba siendo el esperado.

3. Las opciones de depuración dentro de la pestaña de código (source) me ha permitido introducir puntos de interrupción con los que detener el flujo para comprobar en cualquier instante si los valores de las variables eran correctos.

## **OneNote**

Se trata de un programa desarrollado por Microsoft, cuya funcionalidad principal es facilitar la toma de notas y recopilación de información.

OneNote ha sido el software que he usado para ir tomando notas de mi trabajo semanal, de esta forma, he podido llevar una buena planificación.

## **Skype**

Skype es un software que permite que todo el mundo se pueda comunicar desde cualquier lugar, ya sea a través de mensajes, llamadas o videollamadas desde cualquier parte del mundo gracias utilizando el protocolo Voice Over IP .

En Aticsoft el uso de este software es una herramienta crucial, puesto que es la forma que tienen todos los compañeros de comunicarse entre ellos, así como exponer las distintas dudas o problemas que van surgiendo día a día en un grupo donde todos pueden aportar una solución o idea adecuada.

En mi caso, he utilizado Skype para comentar con mi superior los avances y dudas acerca del proyecto ya sea a través del chat o de videollamadas.

Esta herramienta me ha venido muy bien ya que la mitad de mi estancia en prácticas la tuve que realizar teletrabajando y no podía acudir presencialmente a la empresa.

## **Xampp**

XAMPP es un paquete de software libre que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl [\[12\]](#).

Cuando te descargas XAMPP viene con 5 módulos de los cuales arrancarás únicamente los que sean necesarios para tu aplicación. En mi caso, para poder realizar mi proyecto he tenido que arrancar Apache, MySQL y Mercury.

## **Apache**

Apache es un servidor web HTTP abierto y gratuito que está desarrollado y mantenido por la Apache Software Foundation.

He utilizado Apache en mi proyecto como servidor, para poder cargar mi página web localmente, al finalizar la página web y ver que todo iba correctamente trasladé todo el directorio al servidor de la empresa.

## **Mercury**

Es un servidor de correo compatible con los estándares desarrollado por David Harris y que está incluido en el software de XAMPP.

He utilizado *Mercury* [\[13\]](#) para poder enviar correos con el contenido del mensaje ( el mensaje estará formado por las plantillas que haya seleccionado) desde el servidor localhost hasta el exterior. Para ello, he tenido que configurar el mail server de XAMPP.

## **PHPMyAdmin**

Es una herramienta gratis y de código abierto, escrita en PHP, y que sirve para administrar bases de datos MySQL a través de una página web.

En mi proyecto he utilizado PHPMyAdmin para crear la base de datos así como sus tablas, que en mi caso contendrán la información de las plantillas (cabeceras, cuerpos y pies de páginas).

## **Filezilla**

En sus versiones cliente o servidor, ambiente multiplataforma,3 de código abierto y software libre, licenciado bajo GPLv2. Soporta los protocolos FTP, SFTP y FTP sobre SSL/TLS (FTPS) [\[14\]](#).

Una vez estaba todo el proyecto finalizado, he utilizado Filezilla para importar el proyecto desde mi directorio local al servidor de la empresa.

## MySQL-Front

Es un software libre y de código abierto que permite conectarse a servidores MySQL (y sus derivaciones como MariaDB y Percona Server), así como Microsoft SQL Server y PostgreSQL [\[15\]](#).

He utilizado MySQL-Front para importar la base de datos junto con sus tablas y contenido desde PHPMyAdmin de mi servidor local al servidor de la empresa.

## Draw.io

Draw.io es una herramienta que nos permite elaborar diagramas en línea sin necesidad de instalar absolutamente nada en nuestro PC [\[16\]](#).

He utilizado esta herramienta para elaborar el diagrama de casos de uso.

De todas las herramientas explicadas anteriormente, tanto Filezilla como Eclipse y MySQL-Front han sido proporcionadas por parte de la empresa. Por mi parte, el uso del resto de herramientas ha sido indispensable para el resultado final del proyecto.

## 2.4. Análisis de riesgos

La parte del análisis de riesgos es muy importante, ya que aquí es donde se estudian las posibles amenazas y eventos no deseados que pueden surgir a lo largo del proyecto. A continuación voy a listar los riesgos a los que se se ha enfrentado este proyecto:

- ❑ Cambio en el modo de trabajo y en el calendario. Al inicio del proyecto se trabajaba presencialmente en la empresa, pero dada la situación inesperada del COVID-19, se tuvo que cambiar el modo en que trabajábamos, pasando a una modalidad de teletrabajo. Paralelamente, las fechas implicadas en este, se fueron retrasando.
- ❑ Problemas de conexión con el servidor. En un momento determinado durante el desarrollo del proyecto, el servidor que me había proporcionado la empresa bloqueó el acceso de mi usuario. Ante esta situación, durante el tiempo que tardaron en restaurar el servidor, decidí instalarme un servidor local (XAMPP) con el que poder avanzar.
- ❑ Cambios en el proyecto. Se modificó el proyecto inicial para que fuese un módulo de envío de emails personalizados, ya que disponíamos de muchísimas horas menos de desarrollo tras las circunstancias generadas por la pandemia descritas anteriormente.

- ❑ Poca experiencia en algunas de las herramientas y lenguajes empleados. La implementación del proyecto requiere de tecnologías tanto del lado del servidor como del lado del cliente, por lo que tuve que aprender en poco tiempo a desenvolverme en cada lenguaje, hecho que se repitió con algunas de las herramientas empleadas.

## 2.5. Planificación

En la fase de planificación, primordialmente repasamos a fondo el proyecto, es decir, definimos una idea clara de los objetivos y el alcance para poder entender bien cuales son las necesidades a suplir, las tecnologías que se usarán y la forma de llevarlo a cabo.

Una vez esto está claro, es necesario realizar una planificación en la que se divida el proyecto en tareas o incluso subtareas, cada una de las cuales podrá depender de la anterior y necesitará de sus predecesoras. Al lado de cada tarea también hay que indicar el tiempo que se tiene pensado invertir, para así al final calcular el total de horas y ver se que cumplen las horas que se habían propuesto para la realización del proyecto.

Luego tenemos la parte de diseño e implementación (capítulos 4 y 5), donde se especificarán las entidades, se desarrollarán los casos de uso y se procederá a implementar el código de la aplicación.

### 2.5.1. Planificación inicial

En la siguiente tabla, se puede ver la planificación inicial del proyecto, con sus tareas y subtareas, y junto con la duración de cada una.

Inicialmente, tenía una duración de 300 horas. Pero finalmente pasaron a ser 240. Este decremento de horas podremos verlo más adelante en la planificación final.

Mi planificación inicial está dividida en un total de 5 fases, cada una, con sus respectivas tareas.

La tabla comienza con el inicio del proyecto, esto fué el 13 de Febrero de 2020, y finalizará el 22 de Mayo de 2020.

	Tarea	Duración	Fecha inicio	Fecha fin	Predecesoras
1	Inicio	24h			

1.1	Definir el proyecto con el tutor y supervisor	6h	13/02/2020	14/02/2020	
1.2	Identificar objetivos y alcance del proyecto	8h	14/02/2020	17/02/2020	1.1
1.3	Definir método de trabajo y tecnologías	5h	17/02/2020	18/02/2020	1.2
1.4	Definir las reuniones periódicas	5h	18/02/2020	19/02/2020	1.3
<b>2</b>	<b>Documentación</b>	<b>9h</b>			
2.1	Investigación funcionalidades a desarrollar	9h	19/02/2020	21/02/2020	1.4
<b>3</b>	<b>Planificación</b>	<b>38h</b>			
3.1	Definición de requisitos	8h	21/02/2020	24/02/2020	2.1
3.2	Diseño diagrama casos de uso	7h	24/02/2020	25/02/2020	3.1
3.3	Revisión información	10h	25/02/2020	26/02/2020	3.2
3.4	Diseño interfaz	8h	26/02/2020	02/03/2020	3.3
3.5	Diseño base de datos	5h	02/03/2020	03/02/2020	3.4
<b>4</b>	<b>Implementación</b>	<b>211h</b>			
4.1	Estudio PHP	20h	03/03/2020	09/03/2020	3.5
4.2	Estudio tecnología Ajax y lenguaje JavaScript	20h	09/03/2020	13/03/2020	4.1
4.3	Implementación del código	139h	23/03/2020	08/05/2020	4.2
4.4	Pruebas	12h	11/05/2020	13/04/2020	4.3
4.5	Corrección de errores	15h	13/04/2020	18/05/2020	4.4
<b>5</b>	<b>Mejoras</b>	<b>18h</b>			
5.1	Diseño de nuevas funcionalidades	6h	19/05/2020	20/05/2020	4.5
5.2	Estudio y elección de las funcionalidades	6h	20/05/2020	21/05/2020	5.1
5.3	Implementación de las funcionalidades escogidas	6h	21/05/2020	22/05/2020	5.2

**Tabla 1:** Planificación inicial del proyecto.

A continuación expongo más detalladamente los requisitos necesarios para la implementación del código junto a las horas dedicadas a cada una de las tareas.

1. Generación de la estructura base de los archivos html y css del proyecto: *35 horas*.
2. Generación de la base de datos y sus tablas: *10 horas*.
3. Generación de los archivos de *backend* para la conexión y obtención de datos de las tablas anteriormente creadas: *10 horas*.

4. *Generación de los archivos de almacenamiento de datos en base de datos y de las funciones de envío de correos masivos: 30 horas.*
5. *Generación de los estilos finales y depuración de envío de emails: 30 horas.*
6. *Repaso conjunto de todo el funcionamiento: 24 horas.*

De esta forma sumamos las 139 horas dedicadas a la implementación del código.

## 2.5.2. Planificación final

En la siguiente tabla, se puede ver la planificación final del proyecto, es decir, la que se llevó a cabo finalmente. Como había mencionado anteriormente, la planificación inicial del proyecto tuvo que ser modificada. Una de las razones fue que tuvo que reducirse la duración del proyecto a causa de la cuarentena, puesto que habían algunos alumnos que tenían sus prácticas paralizadas.

A estos alumnos, se les dijo que cuando volviesen a realizar las prácticas, estas se darían por finalizadas haciendo el 80% de horas del total (240h).

Al haber tenido las prácticas paralizadas durante la cuarentena, a partir de la mitad de marzo no pude realizar nada, por tanto, las fechas de la planificación inicial tampoco servían (únicamente sirvieron las del primer mes). Con esto, el contrato inicial sirvió solamente para las fechas que comprenden desde el 13 de febrero al 17 de marzo de 2020. Tras esta fecha, se firmó otro contrato para la cancelación de las prácticas, de esta forma, solo se tenían en cuenta las horas realizadas hasta ese día.

A finales de junio, cuando la tutora me comentó que ya era posible volver, se volvió a firmar un contrato donde estarían las 98 horas restantes repartidas en 3 semanas. El proyecto empezaría el 13 de Julio de 2020 y finalizará el 31 de Julio de 2020.

Con respecto al proyecto, también hubieron unas series de modificaciones. Al tener 60 horas menos para poder implementar el proyecto final, variaron algunas secciones como la eliminación del proceso de envío de mensajes automatizados que se iba a realizar en un principio. Por tanto, se realizó el mismo proyecto pero quitando esa funcionalidad.

Las horas dedicadas al estudio de los lenguajes y tecnologías también se vieron modificadas ya que se requirieron más horas de las que se habían pensado. A parte, también tuve que dedicarle horas a informarme acerca de Google devTools ya que a lo largo de la implementación del código le di mucho uso.

La propuesta de mejoras que estaba en la planificación inicial, se realizó como se había planeado y al ver que el proyecto fue acabado 2 días antes de la fecha final, se decidieron añadir un par de mejoras, como era el caso de tener una opción de visión de pantalla completa y otra opción de visión de pantalla móvil. Esta última, bastante más complicada pero que al final, se consiguieron sacar las dos mejoras adelante.



	Tarea	Duración	Fecha inicio	Fecha fin	Predecesoras
<b>1</b>	<b>Inicio</b>	<b>24h</b>			
1.1	Definir el proyecto con el tutor y supervisor	6h	13/02/2020	14/02/2020	
1.2	Identificar objetivos y alcance del proyecto	8h	14/02/2020	17/02/2020	1.1
1.3	Definir método de trabajo y tecnologías	5h	17/02/2020	18/02/2020	1.2
1.4	Definir las reuniones periódicas	5h	18/02/2020	19/02/2020	1.3
<b>2</b>	<b>Documentación</b>	<b>9h</b>			
2.1	Investigación funcionalidades a desarrollar	9h	19/02/2020	21/02/2020	1.4
<b>3</b>	<b>Planificación</b>	<b>38h</b>			
3.1	Definición de requisitos	8h	21/02/2020	24/02/2020	2.1
3.2	Diseño diagrama casos de uso	7h	24/02/2020	25/02/2020	3.1
3.3	Revisión información	10h	25/02/2020	26/02/2020	3.2
3.4	Diseño interfaz	8h	26/02/2020	02/03/2020	3.3
3.5	Diseño base de datos	5h	02/03/2020	03/02/2020	3.4
<b>4</b>	<b>Implementación</b>				
4.1	Estudio PHP	20h	03/03/2020	09/03/2020	3.5
4.2	Estudio tecnología Ajax y lenguaje JavaScript	25h	09/03/2020	17/03/2020	4.1
4.3	Estudio Chrome devTools	5h	13/07/2020	13/07/2020	4.2
4.4	Implementación del código	74h	14/07/2020	27/07/2020	4.3
4.5	Pruebas	2h	27/07/2020	28/07/2020	4.4
4.6	Corrección de errores	8h	28/07/2020	29/07/2020	4.5
<b>5</b>	<b>Mejoras</b>	<b>14h</b>			
5.1	Diseño de nuevas funcionalidades	4h	29/07/2020	29/07/2020	4.6
5.2	Estudio y elección de las funcionalidades	2h	29/07/2020	30/07/2020	5.1
5.3	Implementación visión pantalla completa	2h	30/07/2020	30/07/2020	5.2
5.4	Implementación visión pantalla móvil	6h	30/07/2020	31/07/2020	5.3

**Tabla 2:** Planificación final del proyecto.

La pila del proyecto que se había estimado con 139 horas se ha tenido que desarrollar en 79 horas a causa de los problemas ocasionados por la pandemia mencionados anteriormente.

1. Generación de la estructura base de los archivos html y css del proyecto: *25 horas*.
2. Generación de la base de datos y sus tablas: *5 horas*.
3. Generación de las plantillas e introducción de estas en la base de datos: *15 horas*.
4. Generación de los archivos de *backend* para la conexión y obtención de datos de las tablas anteriormente creadas: *7 horas*.
5. Interacción desde el cliente con los archivos de backend anteriormente creados para la obtención de datos dinámicos: *15 horas*.
6. Generación de los estilos finales y repaso conjunto de todo el funcionamiento: *12 horas*.

De esta forma sumamos las 79 horas dedicadas a la implementación del código.

## 2.6. Estimación de recursos y costes del proyecto

En esta sección, se va a hablar de los recursos y costes del proyecto, tanto materiales, como humanos.

En cuanto a costes materiales, tenemos los siguientes:

- Portatil ACER (446€).
- Monitor de pantalla HP (159€).
- Teclado y ratón HP (35€).

La suma total de estos costes es de 640€. Teniendo en cuenta la vida útil del hardware, que suele ser entre 6 y 8 años, nos salen dos escenarios a tener en cuenta.

- En el mejor de los casos, sería que el hardware nos llegase a durar el máximo tiempo posible, es decir, 8 años, que son 2920 días. Si contamos el uso que le hemos dado en las 240 horas impartidas (que son 40 días) y teniendo en cuenta que hacía 6 horas diarias de lunes a jueves, sale un total de 8€.

640	—————	2920
X	—————	40
<hr/>		
$X = (40 \cdot 640) / 2920 = 8€$		

**Figura 1:** Cálculo peor caso.

- En el peor de los casos, sería que el hardware nos llegase a durar 6 años, que son 2190 días. Contando las 240 horas impartidas (40 días), sale un total de 11€.

640	—————	2190
X	—————	40
$X = (40 \cdot 640) / 2190 = 11\text{€}$		

**Figura 2:** Cálculo peor caso.

A lo largo de toda mi estancia en prácticas, he tenido que desarrollar el proyecto con el programa eclipse, aunque la descarga de este programa no supondría ningún coste adicional ya que es gratuito.

En cuanto a los costes humanos, vamos a hablar del coste del programador. Para calcularlo, tenemos en cuenta el coste por hora y el número de horas trabajadas. En Aticsoft se estima el precio en unos 10€/hora de desarrollo a los cuales hay que añadir un 20% de costo extra por impuestos. Las horas dedicadas a la realización del proyecto han sido de 240 horas, y este ha sido realizado de forma individual. Teniendo en cuenta eso, si multiplicamos los 10€/hora por las 248 horas que se le han dedicado al proyecto, hacen un total de 2480€. Si a esa cantidad le sumamos el 20% mencionado anteriormente se quedaría en un total de 2976€.

Si al coste de los elementos de hardware le añadimos los costes humanos, este asciende a 2984€ en el mejor de los casos y a 2987€ en el peor de los casos.

A todo esto anterior habría que añadirle los costes indirectos. Dentro de esta sección entrarían los costes mensuales de agua, luz, alquiler del local, factura telefónica y de internet.

Si a los costes totales le añadimos un 20% de los gastos mencionados anteriormente se quedaría un total de 2985,6€ en el mejor de los casos y un total de 2989,2€ en el peor de los casos.

	Precio mejor caso	Precio peor caso
Horas trabajadas por precio media	2976€.	2976€.
Eclipse	Gratuito	Gratuito
Hardware	8€	11€
Costes indirectos	2984€ * 0,2 = 596,8€	2987€ * 0,2 = 597,4€
<b>Coste total</b>	<b>2.572,8€</b>	<b>3.573,4€</b>

**Tabla 3:** Tabla resumen del coste total del proyecto.

## 2.7. Seguimiento del proyecto

Al seguirse la metodología ágil SCRUM para realizar el proyecto, cada dos semanas tiene lugar un sprint diferente. En cada uno de los diferentes sprints, se pueden ver cada una de las diferentes tareas que se llevan a cabo, junto con los progresos que se van haciendo y las dudas que vayan surgiendo.

### 2.7.1. Sprint 1

#### 2.7.1.1. Planificación del Sprint 1

Al ser el primer Sprint, se llevó a cabo una pequeña reunión inicial. Se hablaron de los objetivos del proyecto y de las tecnologías y lenguajes a usar. El tutor, en este caso hacía de Product Owner y Scrum Master.

Éste, comentó que sobre todo se gastaría la tecnología Ajax que se usa a través del lenguaje de programación JavaScript y se comunica con el lenguaje del servidor PHP. Al tener conocimientos muy básicos sobre estas tecnologías, se creyó conveniente dedicar este tiempo a profundizar e informarse para tener una fluidez y control adecuados. Por otro lado, se crearon los diagramas de casos de uso y de actividades del proyecto. De esta forma se tendría más claro cómo interviene el usuario y a la hora de hacer la implementación, las necesidades de éste serían más claras.

Todo esto que hemos comentado, tuvo lugar la primera semana. Durante la segunda semana, me centré en realizar programas y ejemplos de prueba con PHP y usando la tecnología ajax para ver si había entendido los conocimientos teóricos que había adquirido la semana anterior.

#### 2.7.1.2. . Planificación del Sprint 2

En el siguiente Sprint, se vería si era necesario seguir indagando acerca de la tecnología y lenguajes mencionados o si ya podíamos ir avanzando hacia otro punto.

## 2.7.2. Sprint 2

### 2.7.2.1. Planificación del Sprint 2

En este Sprint, al ver que los conocimientos que se habían adquirido acerca de lo mencionado en el Sprint anterior, se pasó al estudio de Chrome devTools. Este estudio tuvo una duración de unos días y a parte también se hicieron varias pruebas para ir viendo cómo funcionaba la herramienta ya que iba a ser muy usada en la fase de la implementación.

En esta fase también se habló acerca de la implementación de la interfaz de la aplicación. Primero, empecé generando la estructura html y css para la base del proyecto. A continuación, comencé a desarrollar el inicio de la aplicación, que se trataba de un menú de opciones donde los lenguajes que utilizamos para crearla fueron html, css y js, junto con sus respectivos frameworks con los que la programación fue más simple.

### 2.7.2.2. Planificación del Sprint 3

En la siguiente reunión se vería si la implementación era correcta.

## 2.7.3. Sprint 3

### 2.7.3.1. Planificación del Sprint 3

Procedemos a corregir pequeños fallos que surgieron en la implementación del css del Sprint anterior, y luego al ver que la implementación anterior era correcta, se siguió con la implementación del código. A continuación creamos los archivos del servidor que manejarían la información contenida en la base de datos.

### 2.7.3.2. Planificación del Sprint 4

En el siguiente Sprint nos vamos a dedicar a ver las funcionalidades que tiene la aplicación hasta ahora y a añadir un par más que teníamos pendiente.

## 2.7.4. Sprint 4

### 2.7.4.1. Planificación del Sprint 4

En este Sprint, me dedico a programar diferentes interfaces para seguir añadiendo funcionalidades a la web.

En primer lugar, permito que cuando el usuario arrastre plantillas a su espacio destinado, se le permita modificar tanto el texto, como las imágenes de las plantillas, únicamente haciendo un click con el botón izquierdo sobre lo que desea editar.

A su vez he programado la opción de enviar mensaje, la cual me permite enviar el mensaje una vez el usuario ha redactado las plantillas o mostrar un mensaje de error si el usuario le ha dado a redactar sin haber seleccionado plantillas previamente.

#### 2.7.4.2. Planificación del Sprint 5

En el siguiente Sprint comprobamos que las implementaciones realizadas funcionan correctamente y a parte le seguiré añadiendo más funcionalidades a nuestra web.

### 2.7.5. Sprint 5

#### 2.7.5.1. Planificación del Sprint 5

Comentamos una serie de errores que habían aparecido en la implementación anterior y los corrijo. Una vez vemos que lo programado hasta ahora funciona correctamente podemos seguir realizando más cosas.

En este Sprint se realizó la implementación mediante la tecnología Ajax, usando JavaScript, de la funcionalidad de drag and drop, lo cual permite al usuario no tener que clicar sobre la plantilla para que esta aparezca en el espacio destinado al mensaje, sino arrastrar la plantilla hasta el, lo cual queda visualmente más bonito. El crear esta funcionalidad fue algo que me costó bastante pero que al final pude sacar adelante. También se crearon un par de interfaces más.

Por último añadí la funcionalidad de cambios de estilos de las plantillas por medio del uso de librerías como *farbtastic*.

#### 2.7.5.2. Planificación del Sprint 6

En el siguiente y último Sprint, se revisarán las funcionalidades implementadas en el Sprint anterior y se procederá a mejorar el código y estudiar si es posible añadir algunas mejoras si el tiempo restante lo permite.

### 2.7.6. Sprint 6

#### 2.7.6.1. Planificación del Sprint 6

En este último Sprint se realizaron las pruebas necesarias para ver que el funcionamiento de la página era el correcto. Salieron un par de errores que se pudieron solucionar en un par de días, y luego también se realizó la implementación de dos mejoras, las

cuales eran permitirle al usuario la opción de tener la vista de la página en pantalla completa y otra opción con vista móvil.

También me dediqué a refactorizar código para que fuese más comprensible, dividirlo en más carpetas e ir separando el código según sus funcionalidades.





## Capítulo 3

# Análisis y diseño del sistema

En este capítulo, voy a hablar del análisis que hay que hacer antes de ponerse con la implementación del sistema. En esta sección se podrá ver la definición de requisitos, el diagrama de casos de uso junto con su descripción y el diagrama de actividades. Más tarde hablaré del diseño de la arquitectura y de la interfaz.

Finalmente, acabaré hablando del diseño de la base de datos que se ha llevado a cabo.

### 3.1. Definición de requisitos

La definición de requisitos es un punto muy importante porque es donde se recogen tanto las necesidades que van a tener los usuarios como las necesidades software que serán necesarias para satisfacer las necesidades de los usuarios.

Una correcta definición de requisitos nos permitirá tener en el siguiente punto una mejor visión de nuestros diagramas.

En las siguientes figuras, se van a nombrar y detallar cuáles son los requisitos necesarios en la aplicación.

Los requisitos que tienen que ver tanto con la parte del usuario como con la del administrador son:

Requisito 01
<b>Identificador:</b> R01
<b>Autor:</b> Laura
<b>Fecha:</b> 05/03/2020
<b>Revisión:</b> Laura
<b>Versión:</b> 1.0
<b>Descripción:</b> El sistema debe permitir tanto al usuario como al administrador poder consultar todas las plantillas.

**Tabla 4 :** Requisito 01.

<b>Requisito 02</b>
<b>Identificador:</b> R02  <b>Autor:</b> Laura  <b>Fecha:</b> 05/03/2020  <b>Revisión:</b> Laura  <b>Versión:</b> 1.0
<b>Descripción:</b> El sistema debe permitir tanto al usuario como al administrador arrastrar las plantillas a su respectivo espacio para posteriormente poder editarlas y escribir el mensaje.

**Tabla 5 :** Requisito 02.

<b>Requisito 03</b>
<b>Identificador:</b> R03  <b>Autor:</b> Laura  <b>Fecha:</b> 05/03/2020  <b>Revisión:</b> Laura  <b>Versión:</b> 1.0
<b>Descripción:</b> El sistema debe permitir tanto al usuario como al administrador poder editar tanto el texto, como las imágenes de las plantillas.

**Tabla 6:** Requisito 03.

<b>Requisito 04</b>
<b>Identificador:</b> R04  <b>Autor:</b> Laura

<b>Fecha:</b> 05/03/2020
<b>Revisión:</b> Laura
<b>Versión:</b> 1.0
<b>Descripción:</b> El sistema debe permitir tanto al usuario como al administrador modificar las tonalidades de las plantillas.

**Tabla 7:** Requisito 04.

<b>Requisito 05</b>
<b>Identificador:</b> R05
<b>Autor:</b> Laura
<b>Fecha:</b> 05/03/2020
<b>Revisión:</b> Laura
<b>Versión:</b> 1.0
<b>Descripción:</b> El sistema debe permitir tanto al usuario como al administrador poder enviar un mensaje.

**Tabla 8 :** Requisito 05.

Los requisitos que tienen que ver únicamente con la parte del administrador son:

<b>Requisito 06</b>
<b>Identificador:</b> R06
<b>Autor:</b> Laura
<b>Fecha:</b> 05/03/2020
<b>Revisión:</b> Laura
<b>Versión:</b> 1.0
<b>Descripción:</b> El sistema debe permitir al

administrador poder crear, editar y eliminar las plantillas desde la base de datos.

**Tabla 9 :** Requisito 06.

## 3.2. Análisis del sistema

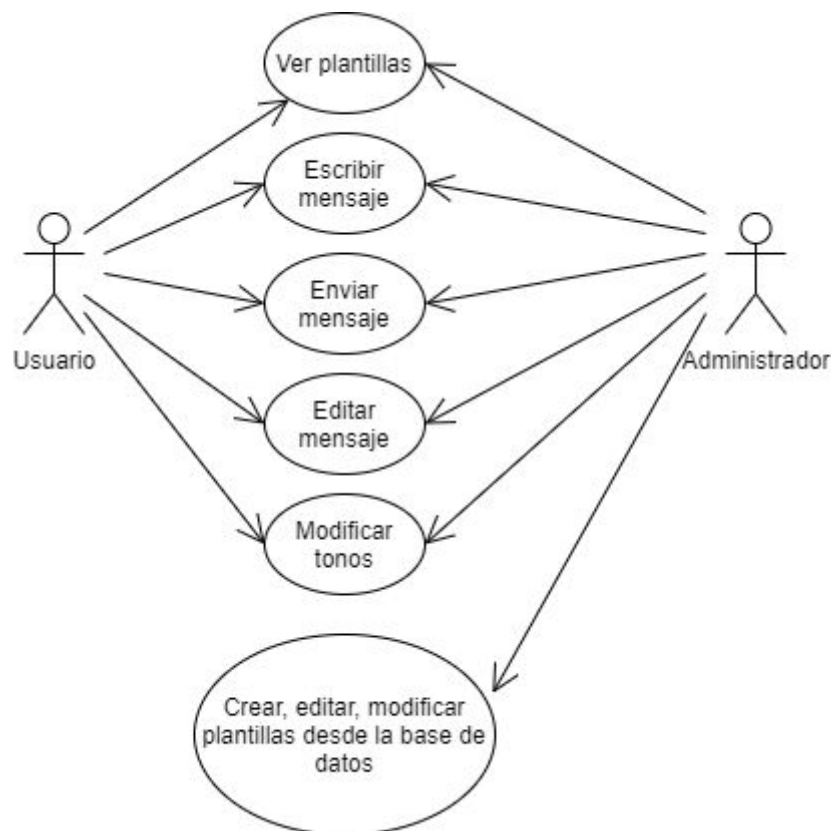
A la hora de realizar el análisis del sistema, voy a empezar mostrando el diagrama de casos de uso de la aplicación y luego mostraré el diagrama de actividades.

El diagrama de casos de uso es necesario para ilustrar los requerimientos del sistema, además, la mejor forma de realizar el análisis del sistema es a través de estos.

### 3.2.1. Diagrama de casos de uso

El diagrama de casos de uso describe las actividades que llevará a cabo alguien a la hora de realizar algún proceso. En nuestro caso, se visualizan las actividades que podrá realizar el usuario usando la aplicación.

El diagrama de casos de uso cuenta con dos actores, estos son los que interaccionan en el sistema. Tenemos el usuario, y el administrador.



**Figura 3:** Diagrama de casos de uso.

En las próximas tablas podremos ver las descripciones de los casos de uso anteriores.

<b>Especificación del caso de uso</b>
<b>Identificador:</b> CU01
<b>Nombre:</b> <i>Consultar plantillas</i>
<b>Autores:</b> Laura Marín Serrano
<b>Versión:</b> 1.0
<b>Descripción:</b> El sistema tiene que mostrar al usuario todas las plantillas existentes.
<b>Precondición:</b> Haber insertado las plantillas en la base de datos.
<b>Condición fin con éxito:</b> Las plantillas se muestran correctamente.
<b>Condición fin con fracaso:</b>
<b>Secuencia de pasos:</b> <ol style="list-style-type: none"> <li>1. Entrar en el apartado de Diseño de correo.</li> <li>2. Seleccionar una de las opciones disponibles dentro de esa sección.</li> <li>3. El sistema muestra al usuario las plantillas existentes dentro de la sección.</li> </ol>

**Tabla 10 :** CU01: Consultar plantillas.

<b>Especificación del caso de uso</b>
<b>Identificador:</b> CU02
<b>Nombre:</b> Cambiar imágenes
<b>Autores:</b> Laura Marín Serrano
<b>Versión:</b> 1.0
<b>Descripción:</b> El sistema debe permitir al usuario cambiar las imágenes originales de las plantillas que hay por defecto.
<b>Precondición:</b> Haber seleccionado una plantilla.
<b>Condición fin con éxito:</b> La imagen se cambia correctamente.
<b>Condición fin con fracaso:</b> La imagen no se cambia.

**Secuencia de pasos:**

1. El cliente selecciona la opción.
2. El sistema muestra al cliente las plantillas disponibles con referencia a dicha opción.
3. El cliente arrastra las plantillas que desee a su espacio correspondiente.
4. El cliente clic con el botón izquierdo del ratón en el texto.
5. El sistema le muestra al cliente la carpeta de imágenes del explorador de archivos.
6. El cliente selecciona la imagen que desee.
7. El sistema cambia la imagen por defecto de la plantilla seleccionada, por la que ha escogido el cliente.

**Excepciones:**

1. El sistema no cambia la imagen.
2. La imagen se muestra en un formato erróneo.

**Tabla 11 : CU02: Consultar plantillas.**

<b>Especificación del caso de uso</b>
<b>Identificado:</b> r CU03
<b>Nombre:</b> Editar texto
<b>Autores:</b> Laura Marín Serrano
<b>Versión:</b> 1.0
<b>Descripción:</b> El sistema debe permitir al usuario editar el texto de las plantillas.
<b>Precondición:</b> Haber seleccionado alguna plantilla.
<b>Condición fin con éxito:</b> El texto se edita a conveniencia del usuario.
<b>Condición fin con fracaso:</b> El texto no se puede editar.
<b>Secuencia de pasos:</b> <ol style="list-style-type: none"><li>1. El cliente selecciona la opción.</li><li>2. El sistema muestra al cliente las plantillas disponibles con referencia a dicha opción.</li><li>3. El cliente arrastra las plantillas que desee a su espacio correspondiente.</li><li>4. El cliente clic con el botón izquierdo del ratón en el texto.</li><li>5. El cliente borra el texto original de la plantilla y escribe el mensaje que quiere enviar.</li></ol>

**Excepciones:**

1. Error, el sistema no deja editar el texto por no haber hecho los cambios oportunos en el código para que la plantilla no fuese editable.

**Tabla 12 :** CU03: Consultar plantillas.

Especificación del caso de uso
Identificador CU04
Nombre Modificar tonalidades
Autores Laura Marín Serrano
Versión 1.0
<b>Descripción:</b> El sistema tiene que permitir al usuario cambiar los tonos por defecto de las plantillas.
<b>Precondición:</b> Haber seleccionado una plantilla.
<b>Condición fin con éxito:</b> Los colores se cambian correctamente.
<b>Condición fin con fracaso:</b> Los colores no se cambian.
<b>Secuencia de pasos:</b> <ol style="list-style-type: none"><li>1. El cliente selecciona la opción.</li><li>2. El sistema muestra al cliente las plantillas disponibles con referencia a dicha opción.</li><li>3. El cliente arrastra las plantillas que desee a su espacio correspondiente.</li><li>4. El cliente selecciona la opción de cambiar color.</li><li>5. El sistema muestra al usuario los colores disponibles para realizar el cambio de color</li><li>6. El cliente selecciona el color que desee.</li><li>7. El cliente selecciona la parte de la plantilla en la cual desea cambiar el color.</li><li>8. El sistema cambia el color de la plantilla.</li></ol>
<b>Excepciones:</b> <ol style="list-style-type: none"><li>1. Los colores no se muestran.</li><li>2. Error, no se cambia el color de la plantilla.</li></ol>

**Tabla 13:** CU04: Consultar plantillas.

<b>Especificación del caso de uso</b>
<b>Identificado:</b> r CU05
<b>Nombre:</b> Enviar mensaje
<b>Autores:</b> Laura Marín Serrano
<b>Versión:</b> 1.0
<b>Descripción:</b> El sistema debe permitir al usuario enviar el mensaje.
<b>Precondición:</b> Haber seleccionado alguna plantilla previamente.
<b>Condición fin con éxito:</b> El mensaje se envía con éxito.
<b>Condición fin con fracaso:</b> Error, no se ha podido enviar el mensaje.
<b>Secuencia de pasos:</b> <ol style="list-style-type: none"> <li>9. El cliente selecciona la opción.</li> <li>10. El sistema muestra al cliente las plantillas disponibles con referencia a dicha opción.</li> <li>11. El cliente arrastra las plantillas que desee a su espacio correspondiente.</li> <li>12. El cliente edita el mensaje como lo desee.</li> <li>13. El cliente selecciona la opción del menú de enviar mensaje.</li> <li>14. El cliente escribe el asunto del mensaje el cual es opcional y el correo del destinatario el cual es obligatorio.</li> <li>15. El sistema envía el mensaje redactado al destinatario.</li> <li>16. El sistema indica que el mensaje ha sido enviado correctamente.</li> </ol>
<b>Excepciones:</b> <ol style="list-style-type: none"> <li>1. El sistema muestra un error si el mensaje no ha podido ser enviado.</li> <li>2. Si el cliente no ha seleccionado ninguna plantilla previamente el mensaje no puede ser enviado.</li> <li>3. Se muestra un mensaje de error cuando el cliente intenta enviar un mensaje pero no rellena los campos correctamente, en este caso, no ha introducido ningún correo de destinatario o el correo que ha introducido no existe.</li> <li>4. Se muestra un mensaje de error si el mensaje no ha podido ser enviado por algún fallo de conexión.</li> </ol>

**Tabla 14:** CU05: Consultar plantillas.



En cuanto al administrador:

<b>Especificación del caso de uso</b>
<b>Identificado:</b> r CU06
<b>Nombre:</b> Crear, editar, borrar plantillas
<b>Autores:</b> Laura Marín Serrano
<b>Versión:</b> 1.0
<b>Descripción:</b> El sistema debe permitir al administrador crear, editar y borrar las plantillas desde la base de datos.
<b>Precondición:</b> Ser el administrador teniendo una cuenta en PHPMyAdmin.
<b>Condición fin con éxito:</b> Las plantillas se crean, editan y se borran con éxito.
<b>Condición fin con fracaso:</b> No se pueden crear, editar o borrar plantillas
<b>Secuencia de pasos:</b> <ol style="list-style-type: none"><li>1. El administrador entra en la base de datos.</li><li>2. El administrador ve las opciones disponibles.</li><li>3. El administrador hace clic sobre el botón de añadir si lo que desea es añadir una nueva plantilla.</li><li>4. El administrador hace clic en editar si lo que desea es editar la plantilla.</li><li>5. El administrador hace clic en eliminar si lo que desea es borrar una plantilla.</li><li>6. El administrador hace clic en guardar cambios.</li><li>7. El sistema guarda las operaciones que ha realizado el administrador.</li></ol>
<b>Excepciones:</b> <ol style="list-style-type: none"><li>1. Error, el sistema no permite a la base de datos ni crear, editar o eliminar las plantillas correctamente, puede deberse tanto a un fallo de conexión como a un fallo ajeno al sistema.</li><li>2. Si no eres administrador, no te deja entrar en la cuenta de PHPMyAdmin, por tanto, no se pueden realizar las operaciones oportunas con las plantillas.</li></ol>

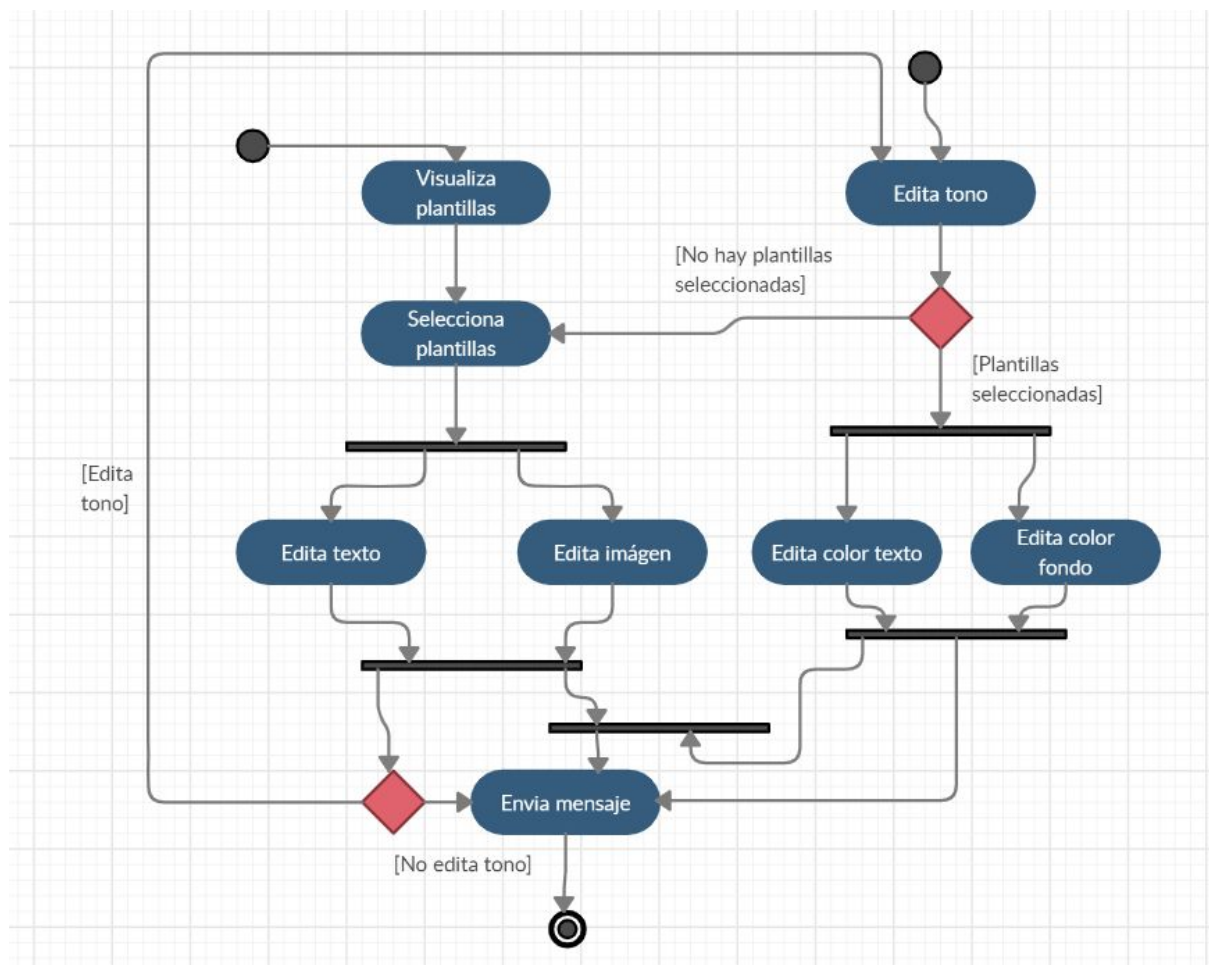
**Tabla 15:** Funciones administrador.

### 3.2.2. Diagrama de actividades

El diagrama de actividades se utiliza para modelar el proceso de un caso de uso. En mi caso lo voy a usar para representar visualmente todas las actividades que es capaz de realizar el usuario en la aplicación.

El paso final sería enviar el mensaje, por tanto, se verán todas las acciones que podrá realizar el usuario hasta llegar a este último caso de uso, el cuál se podrá completar satisfactoriamente o no.

En la siguiente imagen, se observa el diagrama de actividades de mi aplicación.

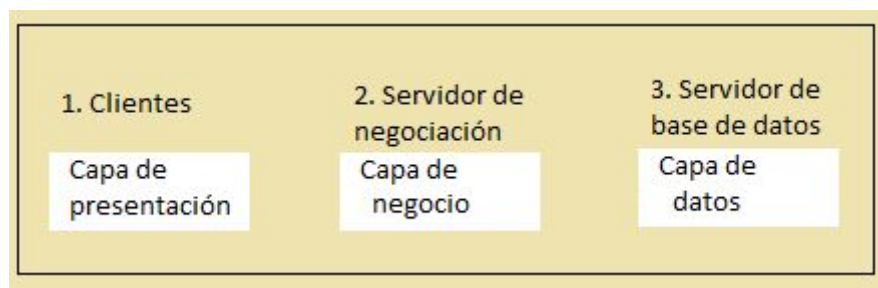


**Figura 4:** Diagrama de actividades.

### 3.3. Diseño de la arquitectura

Durante el desarrollo del proyecto se siguió la estructura de la arquitectura de tres capas. Esta arquitectura se basa en tenerlo todo separado en 3 niveles diferentes, de forma que si surge algún cambio o problema en alguna de las capas, este solo repercute a esa capa en particular y no al resto de capas. Las tres capas que conforman esta arquitectura son las siguientes:

- Capa de presentación.
- Capa de negocio.
- Capa de datos.



**Figura 5:** Arquitectura 3 capas.

En mi proyecto en la capa de presentación estarían tanto los archivos html como las hojas de estilo css y las imágenes. Dentro de la capa de negocio encontraríamos nuestros archivos JavaScript y frameworks de este. Finalmente, dentro de la capa de datos se hallan los archivos PHP encargados de la comunicación con base de datos.

### 3.4. Diseño de la interfaz

En este punto voy a hablar de la interfaz de la aplicación. En la siguiente figura se muestra de forma esquematizada todas las opciones por las cuales puede navegar el cliente.



**Figura 6:** Esquema de navegación del cliente.

El cliente una vez entra en la web, puede ver un menú con 5 opciones:

- La primera opción es *Diseño de correo*. Cuando el cliente selecciona esta opción, se despliega un submenú donde tendrá 4 opciones para cargar las plantillas de una manera estructurada (todas las plantillas, cabeceras, cuerpos o pies de páginas). Al pulsar una de estas opciones podemos apreciar la apertura de un segundo panel lateral que contiene una lista de imágenes que conforman cada una de ellas el diseño por defecto de dicha plantilla.
- La segunda opción del menú es la de *Estilos del correo*. Gracias a esta opción el cliente podrá modificar los estilos de las plantillas relacionados con colores de fondo y letra. Previamente el usuario deberá de haber seleccionado la plantilla que desee editar.

- Si el cliente pulsa sobre *Vista previa*, le aparecerá un submenú con dos opciones. Al seleccionar una de estas podrá obtener una vista previa a modo de pantalla completa o en formato móvil.
- La cuarta opción es la referente al envío de correos. Este solo se enviará si el cliente ha seleccionado alguna plantilla previamente, en caso contrario, saltará una alerta de error ya que no se permite el envío de correos en blanco.
- Por último, seleccionando la opción *Acerca de* podemos encontrar información adicional del desarrollador de la web.

### 3.5. Diseño de la base de datos

Para la base de datos se pensó desde un primer momento que únicamente sería necesaria una tabla donde almacenar las plantillas predefinidas de los correos. El almacenamiento de los datos se llevó a cabo a través de PHPMYAdmin en una tabla llamada "Templates".

A través de él generamos las plantillas diferenciandolas con un campo en 3 grupos: cabeceras, cuerpos y pies de página. De esta forma, nos aseguramos de que podemos consultar por tipo de plantilla si fuera necesario. A su vez, si hubiese que realizar algún mantenimiento posterior, este campo facilitaría mucho manejar las plantillas por grupos.

Más tarde, ante la necesidad de almacenar el proyecto en el servidor de la empresa, utilicé PHPMYAdmin para exportar un archivo con mi base de datos local y a través de MySQL-Front pude importar el archivo sin ningún problema y generar así toda la estructura.

En la siguiente figura, se puede apreciar el contenido de la tabla templates.

id	nombre	contenido	tipo
1	cabecera1	<MEMO>	1
3	cuerpo1	<MEMO>	2
4	pieDepagina1	<MEMO>	3
5	cuerpo2	<MEMO>	2
6	cuerpo3	<MEMO>	2
7	cuerpo4	<MEMO>	2
8	cuerpo5	<MEMO>	2

**Figura 7:** Contenido tabla templates.

Cada una de las plantillas tiene un id asignado, que formará la clave primaria. El nombre que representa a cada una de las plantillas es de tipo varchar. El contenido de las plantillas es

de tipo *text* (visualmente podemos observar la palabra MEMO. MEMO era un tipo de datos utilizado en SQL para almacenar textos sin asignar un tamaño. Con el paso del tiempo, se le cambió el nombre a *text*), y cada una de las plantillas tendrá un contenido específico. Por último, el campo *tipo* está creado para diferenciar los diferentes tipos de plantillas que tenemos, es decir, a las cabeceras, se les ha asignado el entero 1, a los cuerpos, el entero 2, y por último, a los pies de página, el entero 3.

Una vez creamos las plantillas, a través de PHP accedimos a toda su información. De esta forma, pudimos mostrar las plantillas visualmente en nuestro sitio web. Para poder conectarnos a la base de datos y trabajar con esta por medio de consultas, utilizamos la clase PDO de PHP.

PDO (PHP Data Objects) es una extensión que provee una capa de abstracción de acceso a datos para PHP 5, con lo cual se consigue hacer uso de las mismas funciones para hacer consultas y obtener datos de distintos manejadores de bases de datos.

## Capítulo 4

# Implementación

En este apartado, se hablará de la implementación de la aplicación. Se detalla cómo ha sido implementada, se hablará de su estructura, se mostrarán los resultados para posteriormente comentar las dudas que han ido surgiendo a lo largo de la implementación.

### 4.1. Detalles de la implementación

Eclipse, como se ha mencionado anteriormente, es una plataforma de software la cual es utilizada para desarrollar aplicaciones web.

Mi aplicación ha utilizado varias tecnologías para conformar la estructura final.

Por un lado, el esqueleto de la web se formó a través de HTML y su diseño a través de CSS. La funcionalidad de los distintos elementos de la página a nivel de cliente se han implementado usando JavaScript y su framework por excelencia JQuery. Para la comunicación con el servidor sin someter la página a un formulario o a una recarga se ha usado la tecnología Ajax.

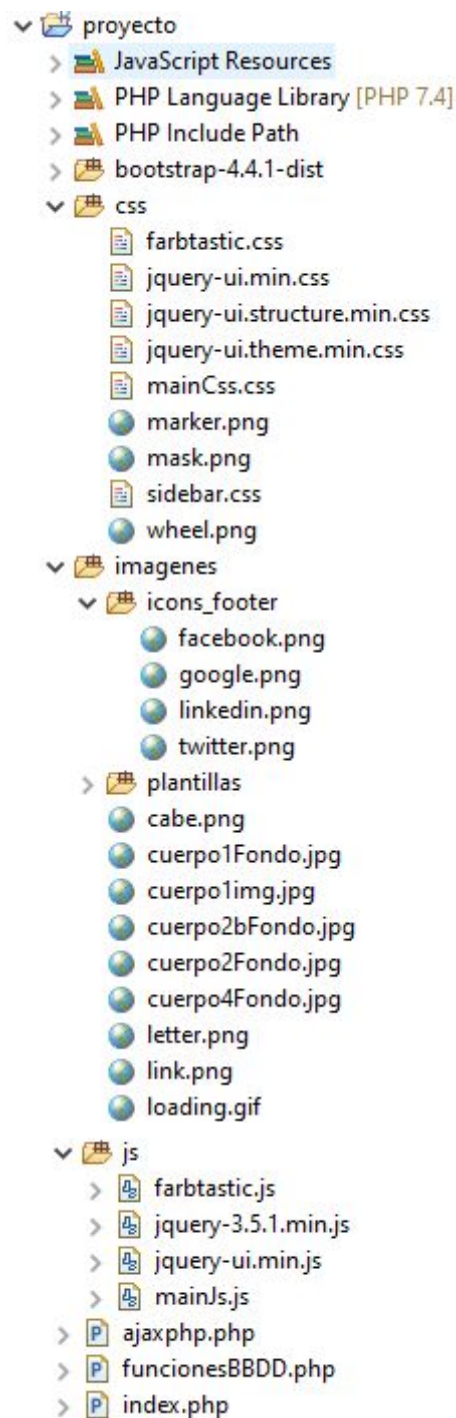
Por otro lado, la funcionalidad que devuelve el contenido de la base de datos y que envía el mensaje se ha implementado en PHP en el lado del servidor. Gracias a PHPMyAdmin y Mysql la implementación de la base de datos ha sido rápida e intuitiva.

Para finalizar, usé frameworks como bootstrap, farbtastic y jquery-ui conseguí modernizar y amenizar la interfaz final de la web.

### 4.2. Estructura de la aplicación

La aplicación, como se ha mencionado en otros puntos, ha sido desarrollada a través de una plataforma llamada Eclipse.

Dentro de él, creé un proyecto, el cual contiene una serie de carpetas con distintos ficheros en cada una de ellas. En la siguiente figura, podremos ver mejor la estructura que ha sido desarrollada a lo largo del proyecto.



**Figura 8:** Estructura del proyecto.

Una vez vista la estructura utilizada para desarrollar la aplicación, voy a explicar lo que contiene cada una de las distintas carpetas y para que se utilizan en el proyecto, ya que únicamente con el aporte visual anterior no es posible entender su funcionalidad.



## Directorio Bootstrap

Carpeta que contiene las librerías de Bootstrap. Podemos encontrar los archivos CSS y los archivos JavaScript relacionados con este framework.

## Directorio CSS

Carpeta que contiene los archivos CSS relacionados con los frameworks de JQuery, JQuery-ui y Farbtastic. A su vez, encontramos:

- *Sidebar.css*: Archivo que contiene todos los estilos referentes a las barras laterales.
- *MainCss.css*: Archivo donde almacenamos todas las reglas de estilos restantes de la web.

## Directorio imágenes

En esta carpeta se almacenan las imágenes que se utilizan para conformar el listado de plantillas. También contiene una imagen con extensión '.gif' que usamos durante el envío de email a modo de icono de carga.

## Directorio JS

En él almacenamos aquellos archivos que contienen código JavaScript. Podemos encontrar la librería de 'Farbtastic.js' responsable del selector del color; las librerías de JQuery, JQuery-ui y en último lugar 'mainJs.js' que contiene todo el código relacionado con la funcionalidad de la página.

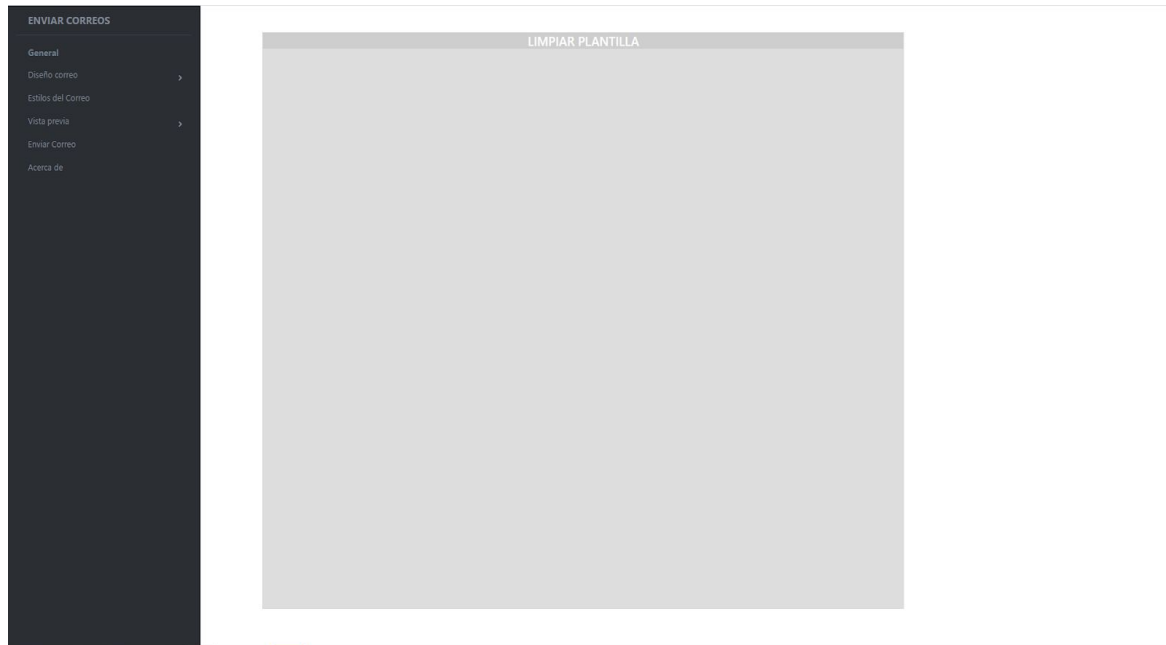
## Raíz

En la raíz del proyecto destacamos estos tres archivos:

- *Index.php*: Archivo que contiene el código html de la web. El resto de elementos html que no se encuentran en este archivo son resultado de una generación dinámica de contenido.
- *Ajaxphp.php*: La importancia de este archivo reside en su funcionalidad, puesto que actúa de puente entre las llamadas que realizamos via Ajax y el archivo 'funcionesBBDD.php' que nos devuelve los resultados.
- *FuncionesBBDD.php*: Por último, en este archivo podemos encontrar las funciones de establecimiento de la conexión y obtención de datos de la base de datos desde el servidor. A su vez, contiene las funciones encargadas del envío de emails.

## 4.3. Resultados de la aplicación

En esta imagen podemos apreciar la carga inicial de la web una vez accede el cliente. Podemos observar que la parte más significativa y la única funcional es la que corresponde con el menú lateral.



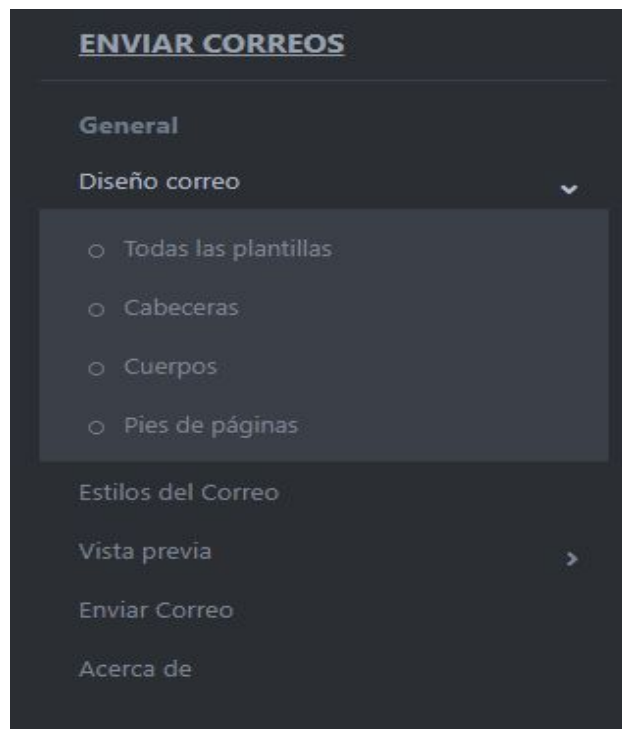
**Figura 9:** Pantalla inicio.

Fijándonos en este panel, observamos las diferentes opciones a las que el cliente tiene acceso.



**Figura 10:** Menú inicio.

Cuando el cliente selecciona la primera opción *Diseño de correo*, se despliega un submenú con una lista de items que al pulsarlos cargan las plantillas vinculadas a esa categoría. A continuación muestro imágenes de este submenú así como las plantillas que estos items cargan en un panel lateral secundario agrupadas por tipo de plantilla.



**Figura 11:** Submenú diseño correo expandido.

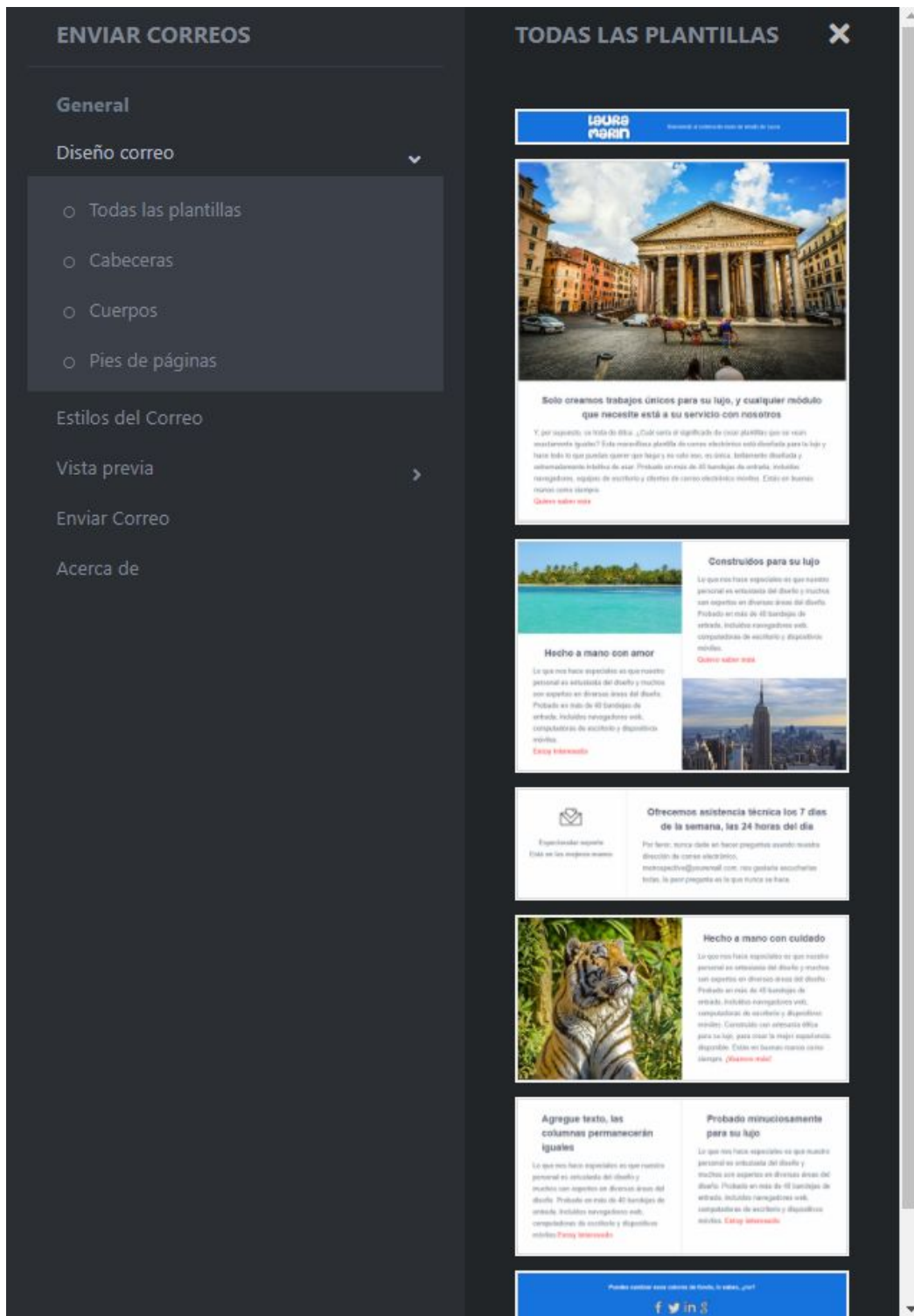
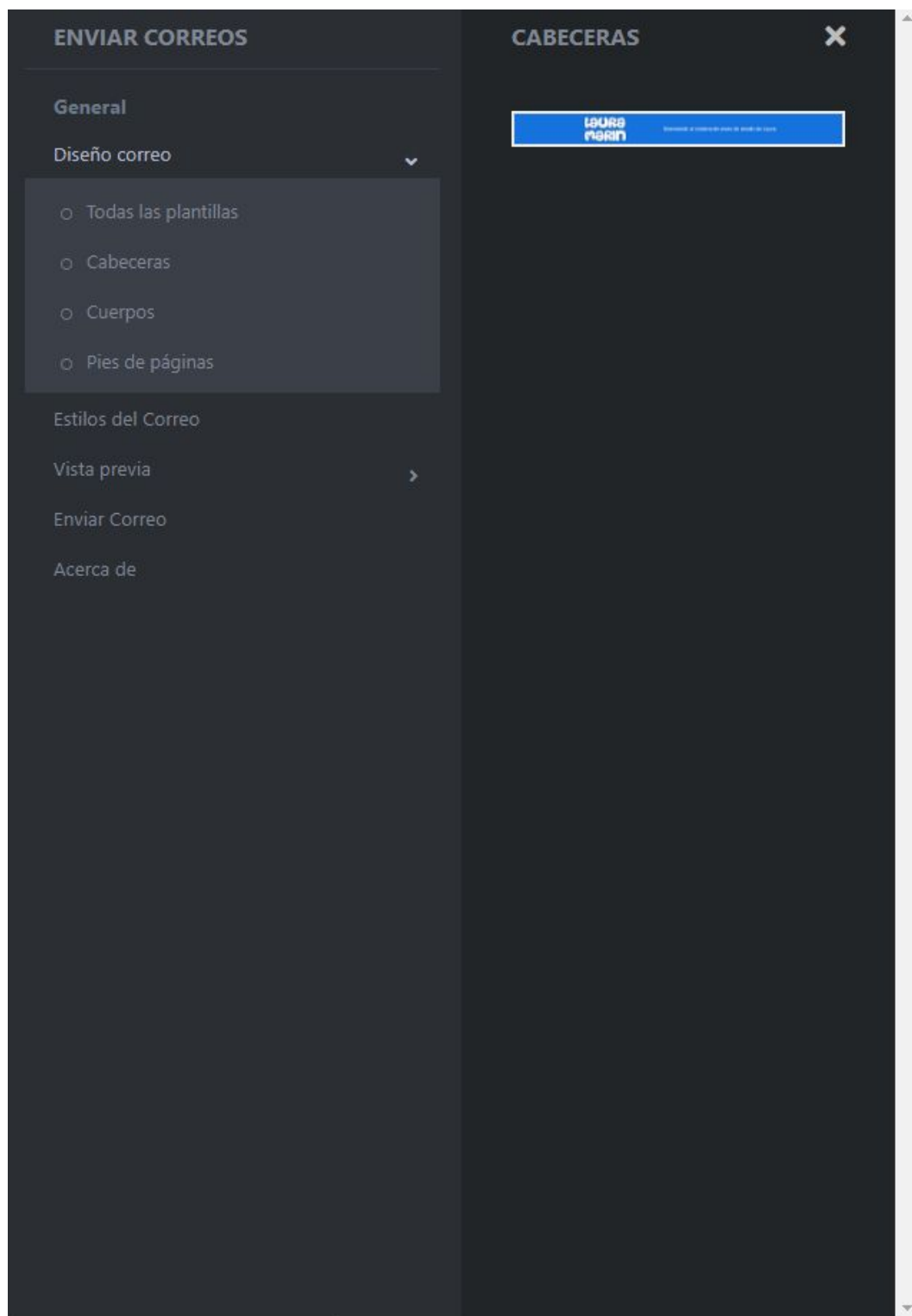


Figura 12: Diseño correo: Todas las plantillas.



**Figura 13:** Diseño correo: Cabeceras.

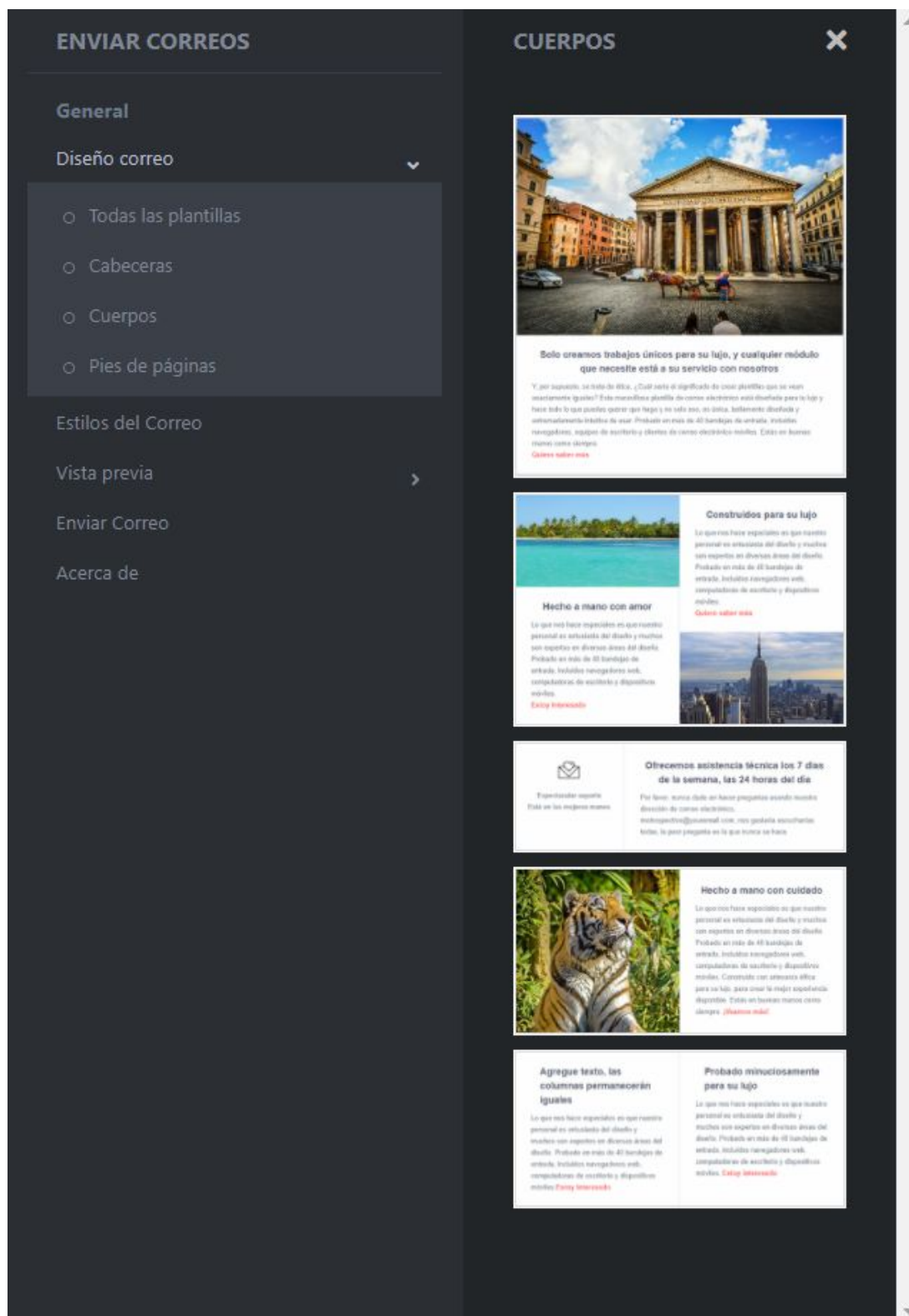
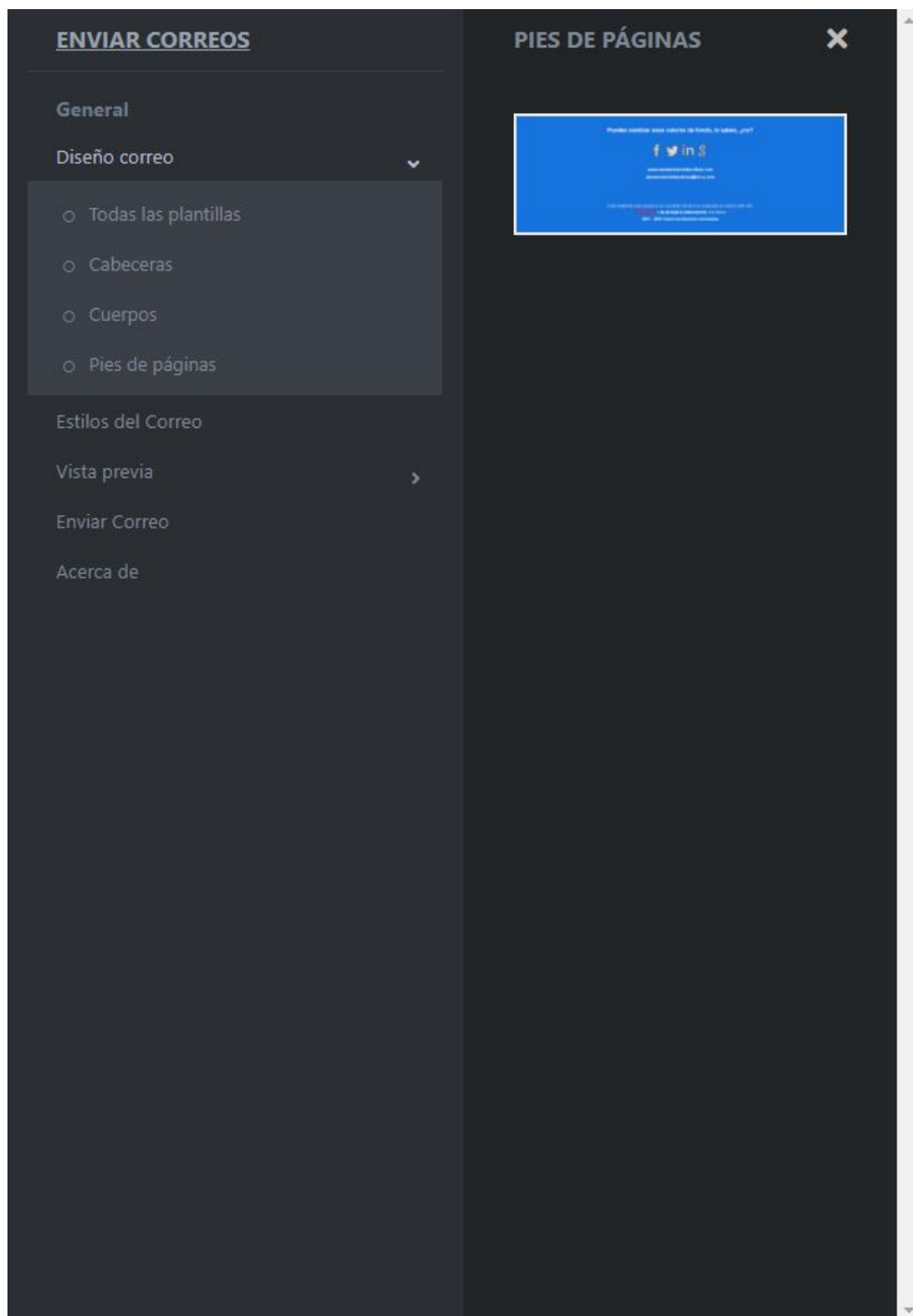


Figura 14: Diseño correo: Cuerpos.



**Figura 15:** Diseño correo: Pies.

Llegados a este punto, tendremos la opción de seleccionar la plantilla o plantillas que deseemos añadir ya sea seleccionándola o arrastrándola al área de edición por medio de la tecnología 'drag and drop'.

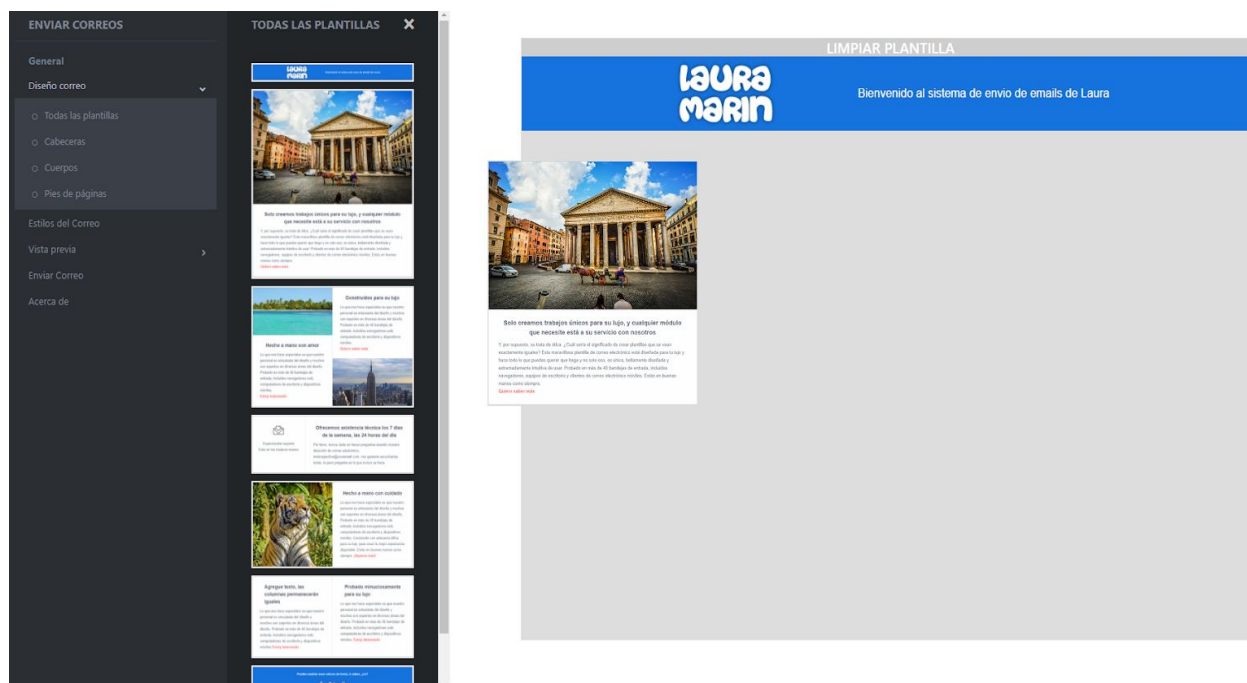
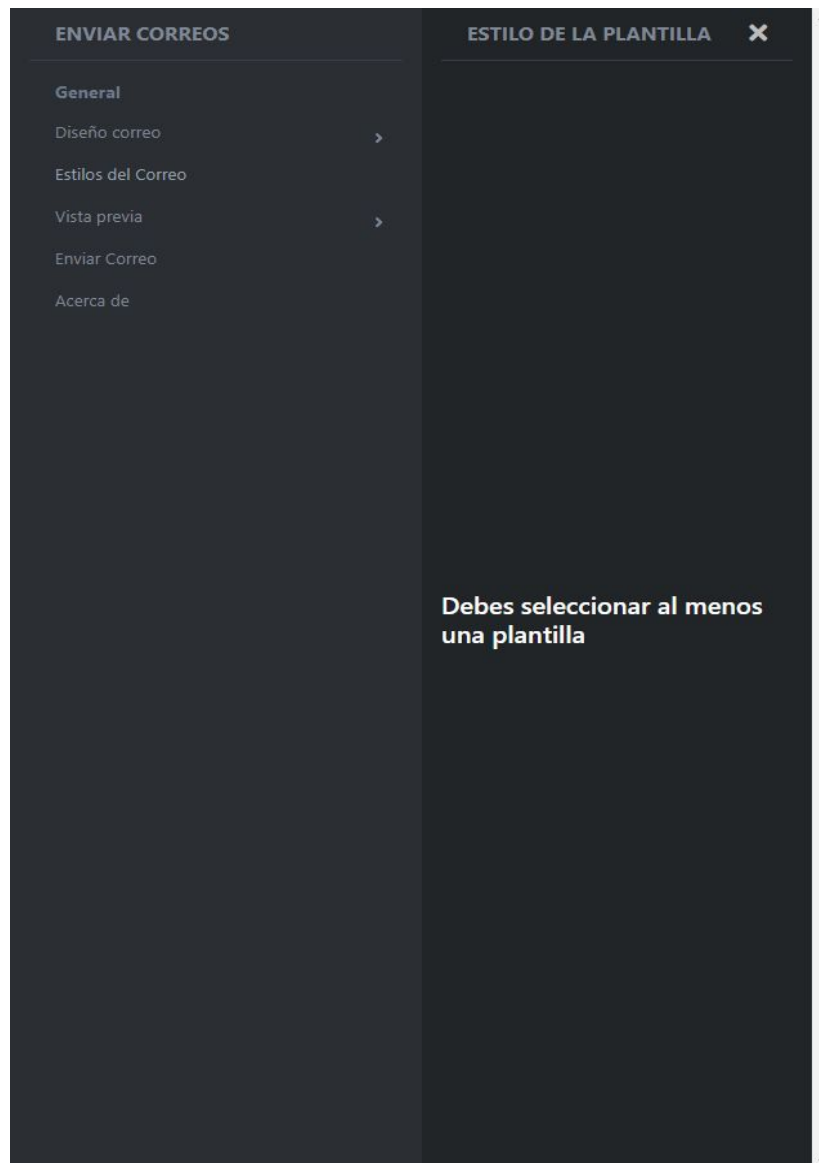


Figura 16: Vista plantilla usando 'drag and drop'.

La siguiente opción que encontramos en el menú es 'Estilos del correo', encargada de mostrar la funcionalidad para cambiar los estilos de fondo y letra de las plantillas. Inicialmente es obligatorio haber seleccionado como mínimo una plantilla del área donde conformamos el correo.





**Figura 17:** Estilos de correo sin selección de plantilla.

Tras seleccionar una plantilla podemos observar que las opciones vinculadas a los textos y fondos están relacionadas con la plantilla escogida, de esta manera cada tipo de plantilla está conformada por opciones diferentes de personalización.

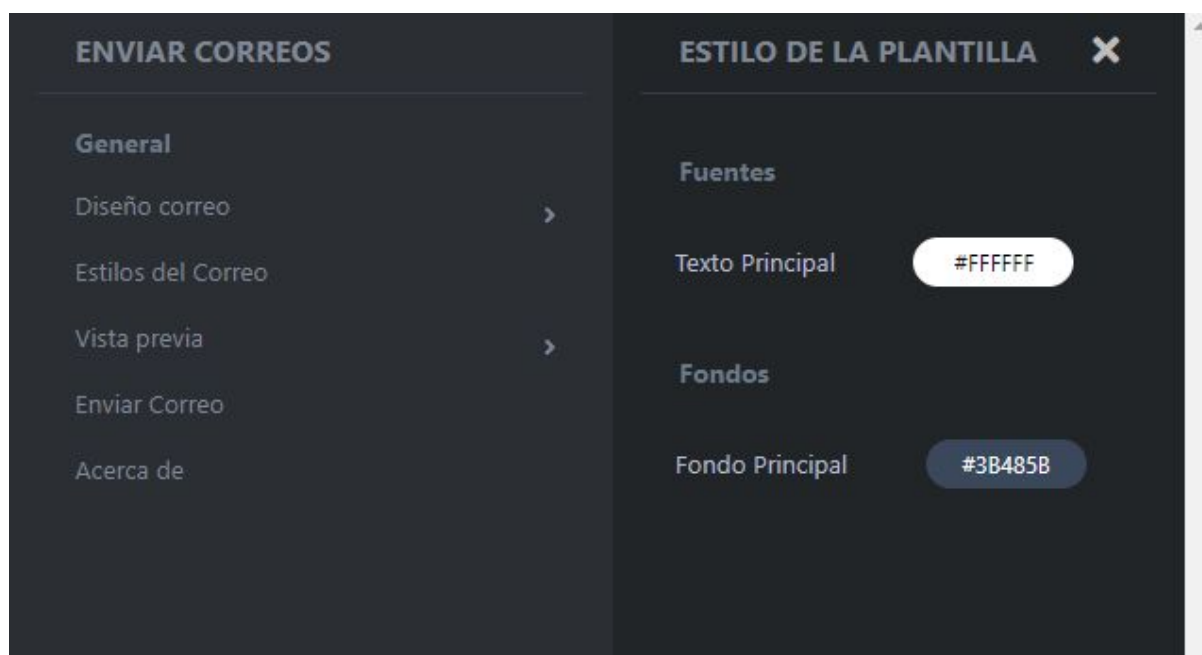


Figura 18: Opciones estilo cabecera.

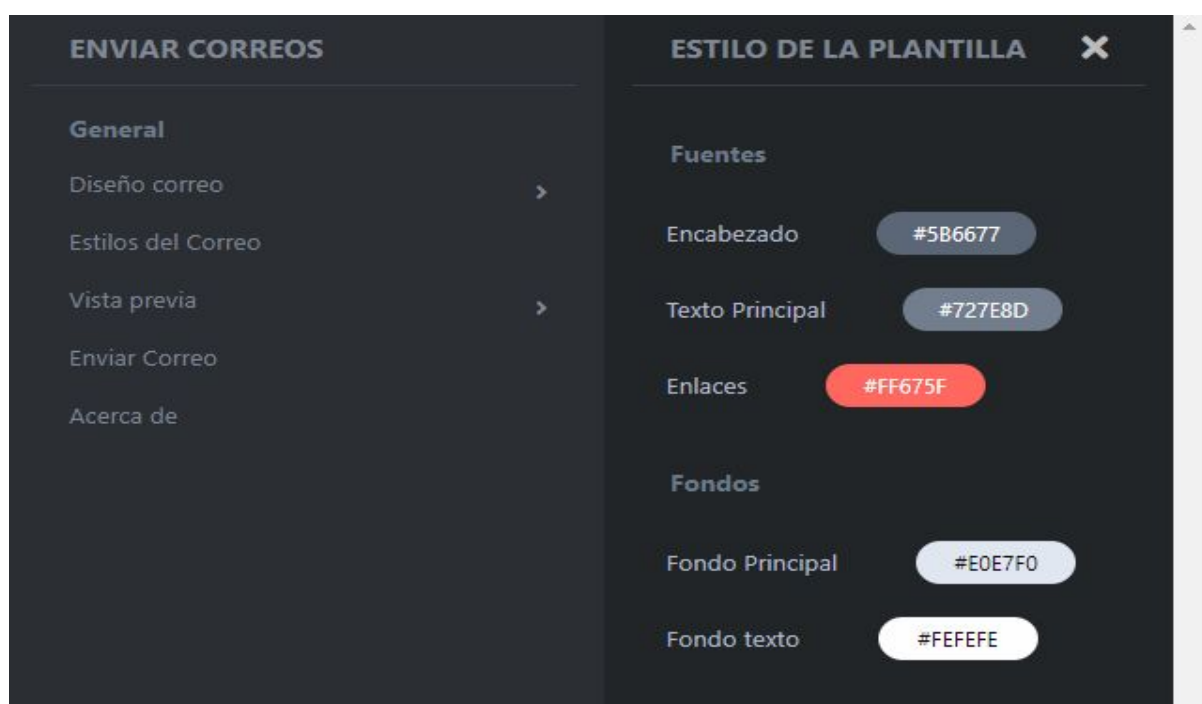
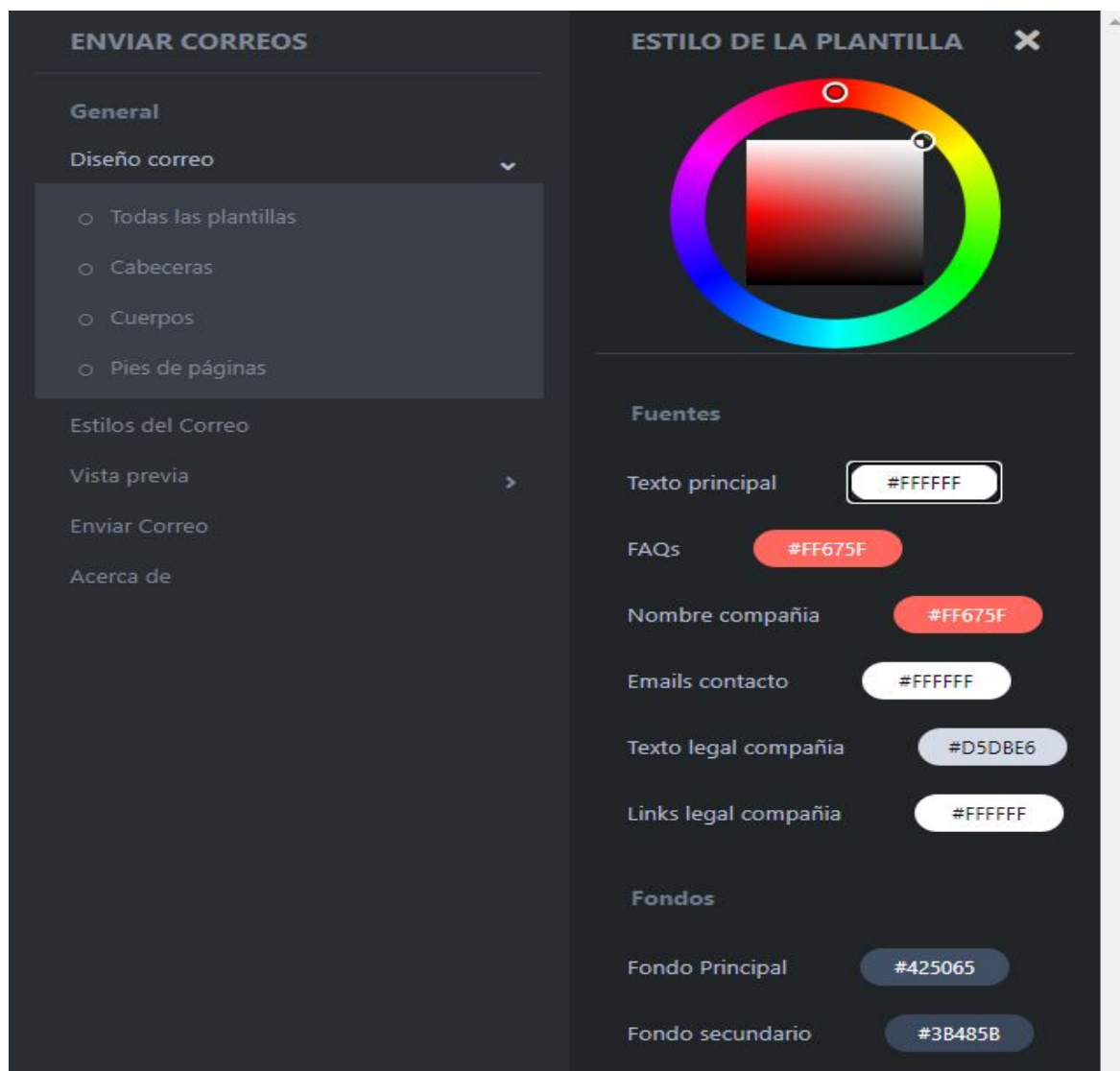


Figura 19: Opciones estilo cuerpo.



**Figura 20:** Opciones estilo pie de página.

Si decidimos modificar algún color simplemente tendremos que seleccionar la opción que deseemos modificar; acto seguido, se mostrará un selector de color como podemos apreciar en la siguiente imagen. Al realizar cambios en el selector de color automáticamente también se cambian tanto en las áreas afectadas por ese ítem así como en el propio ítem.



**Figura 21:** Estilos correo con selector de color visible.

A continuación voy a mostrar un ejemplo de tres plantillas cargadas y más tarde editadas. Podemos apreciar que al dejar el ratón encima de una plantilla, nos aparece un botón rojo que nos permite prescindir de esta. De esta manera no tenemos que descartar todas a través del botón superior 'limpiar plantilla'.

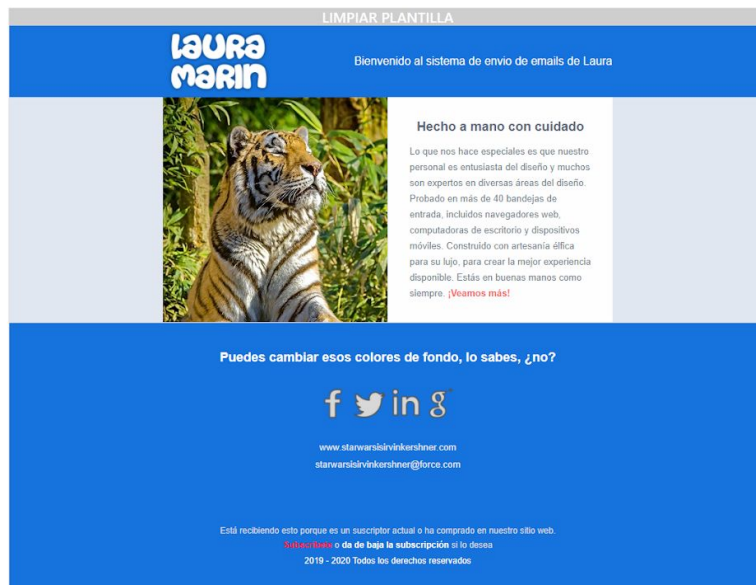
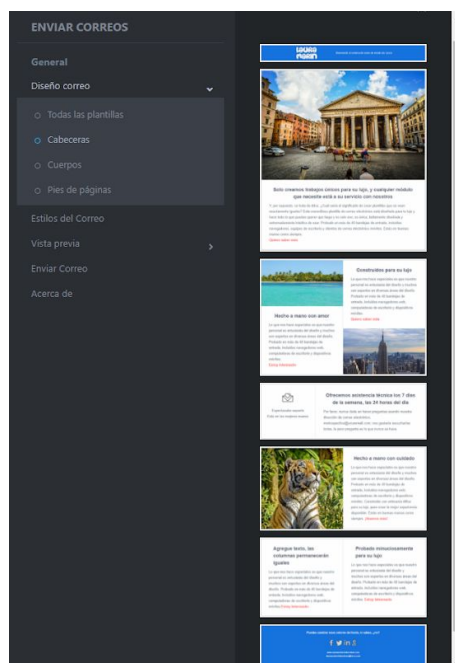


Figura 22: Plantillas originales.

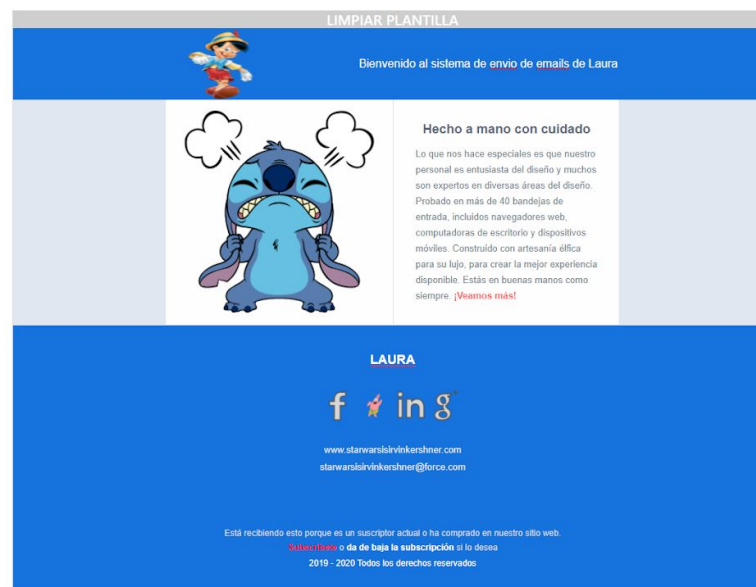
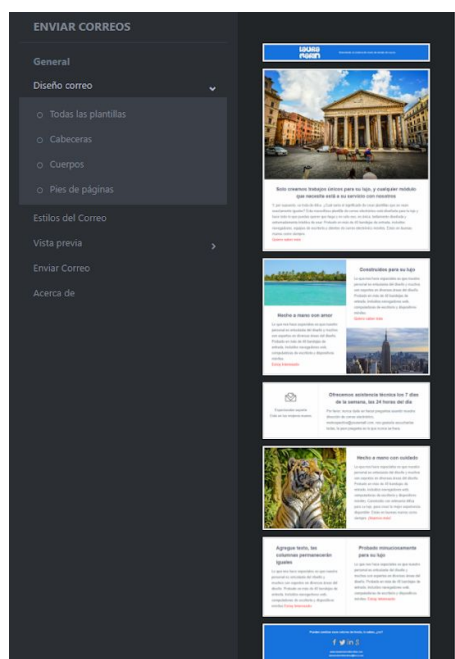
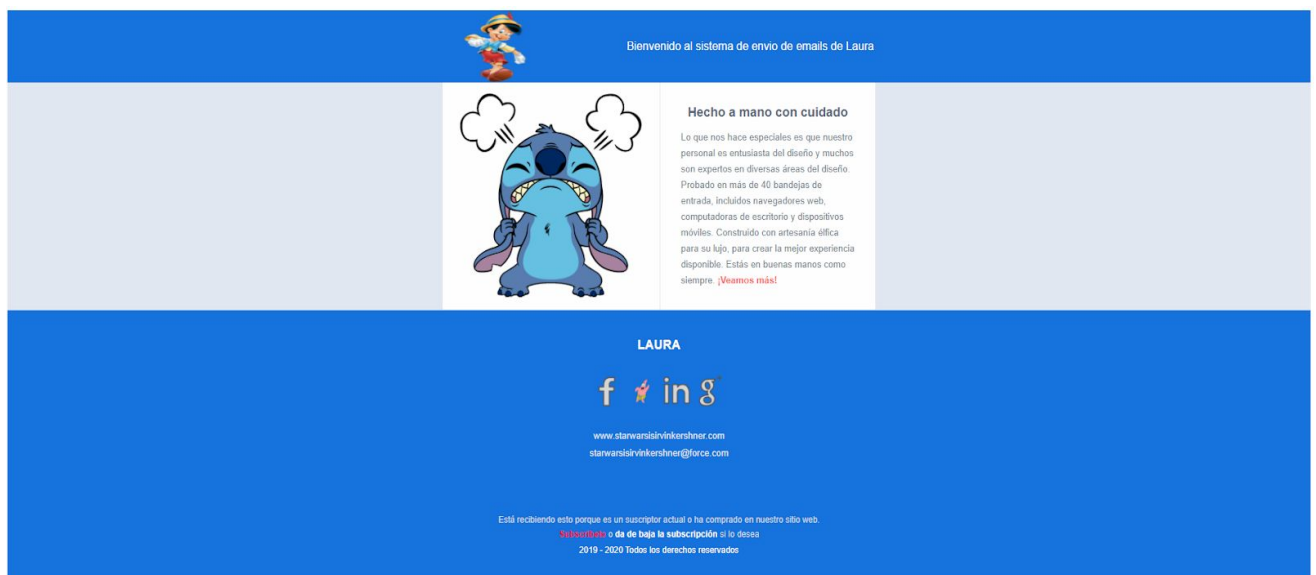


Figura 23: Plantillas editadas.

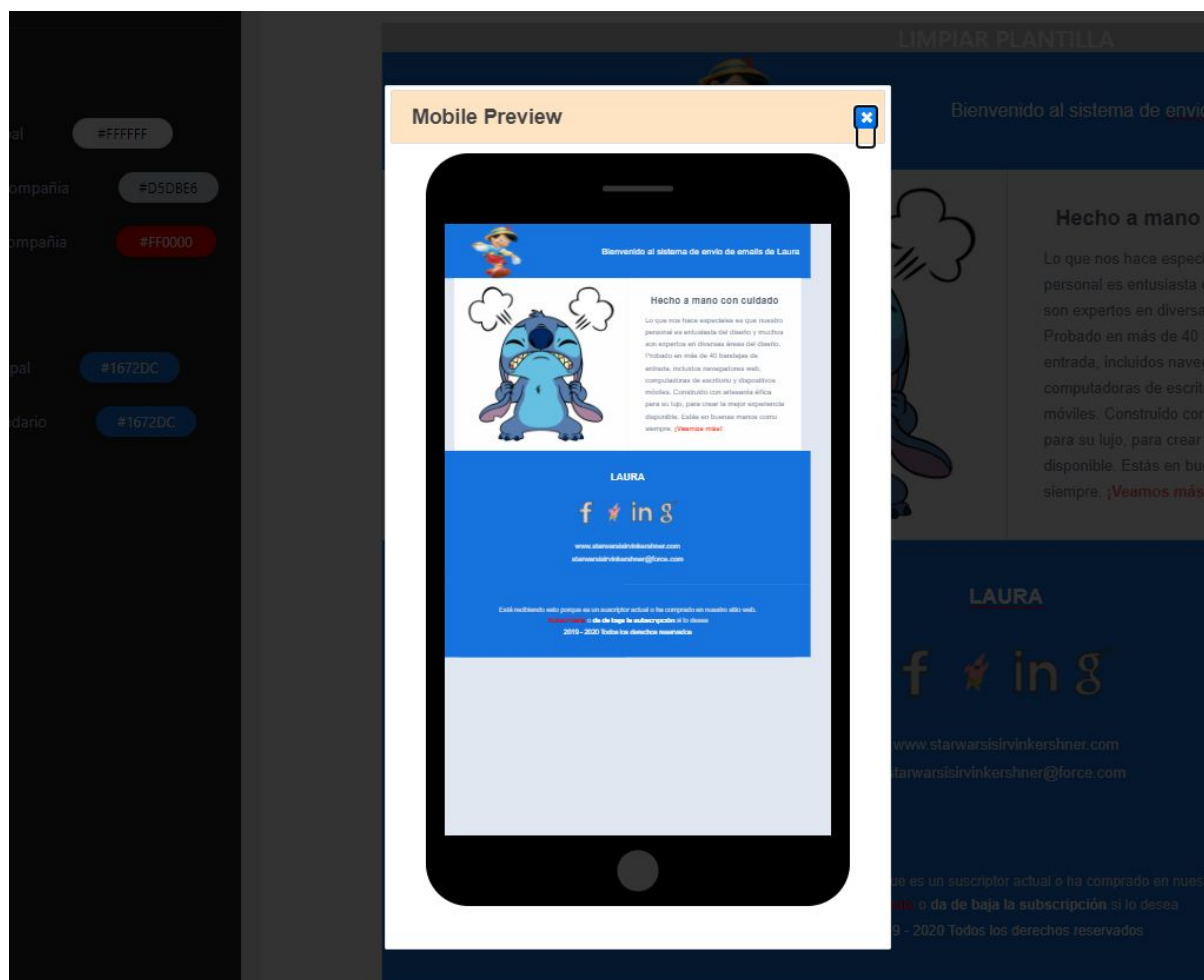
Llegados a este punto, antes de enviar el correo tenemos la posibilidad de visualizar el resultado final en formato pantalla completa o móvil utilizando la opción 'Vista previa' del menú lateral.



**Figura 24:** Vista previa.

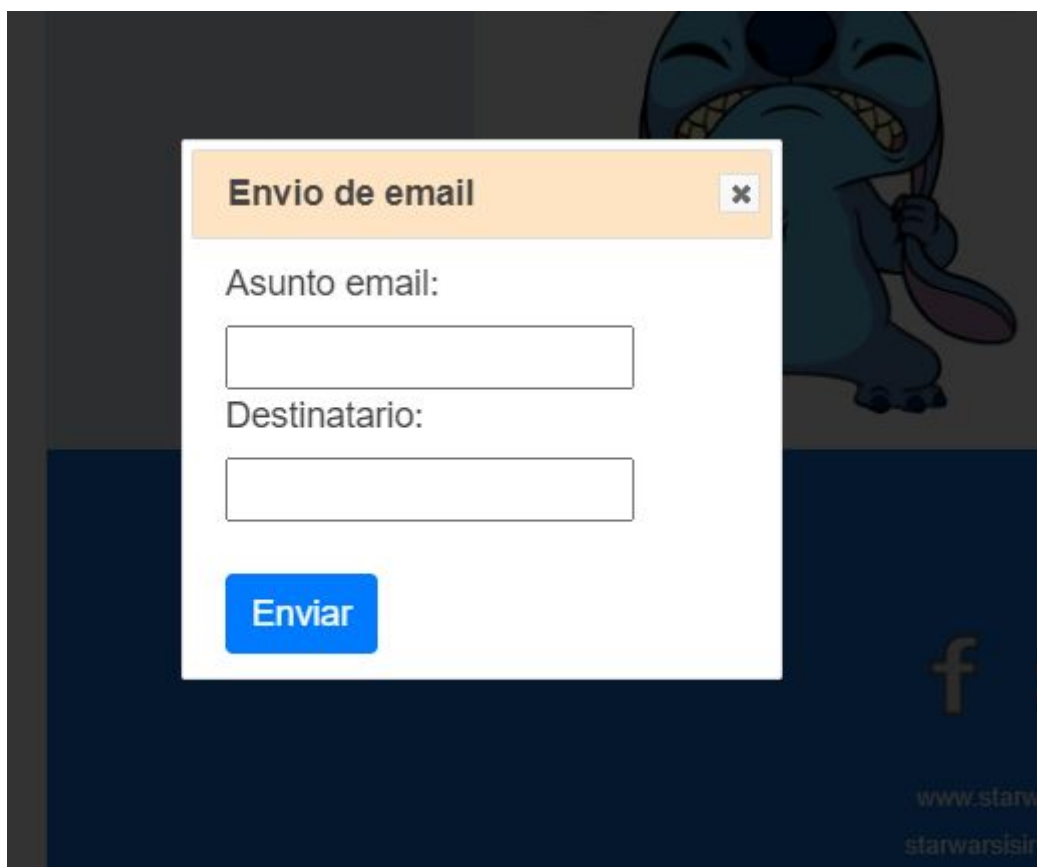


**Figura 25:** Vista previa pantalla completa.



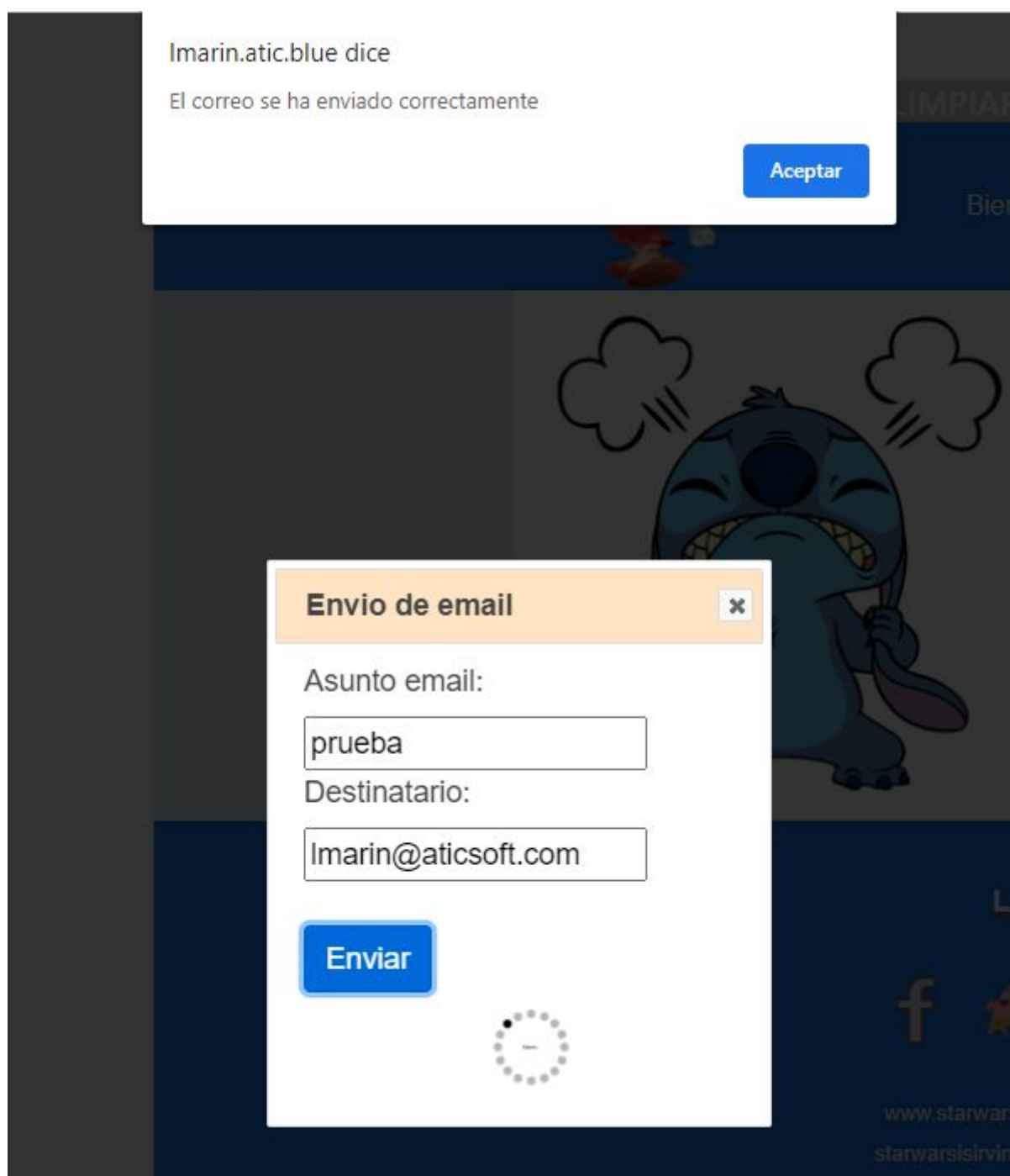
**Figura 26:** Vista previa móvil.

Finalmente hemos llegado al momento de enviar el correo. Para ello seleccionaremos la opción enviar correo de nuestro menú lateral (si no hemos añadido ninguna plantilla nos mostrará un mensaje para avisarnos de que no se pueden enviar mensajes vacíos) y se mostrará un modal donde podremos introducir el asunto del correo y el destinatario (este último es obligatorio, de lo contrario también mostrará un aviso).



**Figura 27:** Modal envío correo.





**Figura 28:** Proceso correcto de envío.

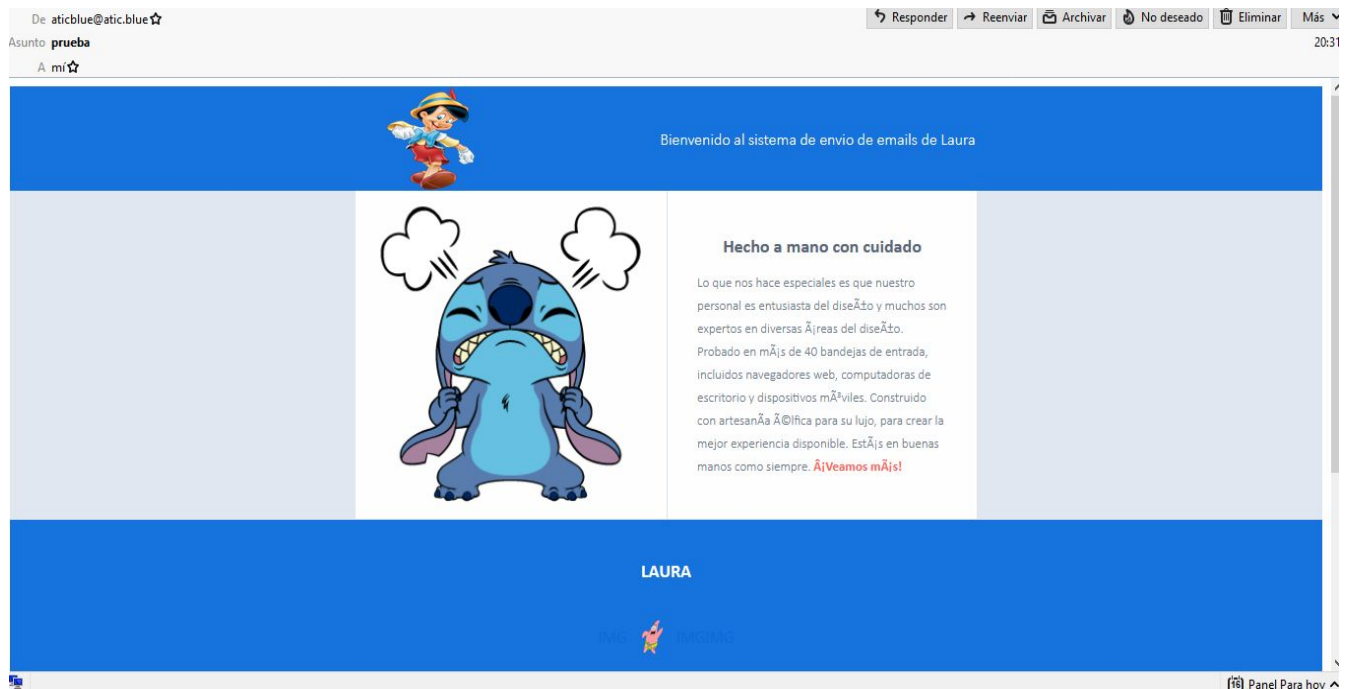


Figura 29: Correo recibido.

## 4.4 Dificultades de implementación

Durante el desarrollo del proyecto, mi mayor dificultad ha sido el cambio entre los diferentes lenguajes que he utilizado. El cambio entre JavaScript y PHP en muchas ocasiones me ha hecho cometer errores básicos de programación.

Respecto al manejo de eventos, sobre todo en elementos que pueden interactuar con el usuario de diferentes maneras ha sido una tarea escabrosa, ya que en varias ocasiones y sin ningún tipo de intencionalidad agregaba múltiples eventos con la misma acción a un único elemento y esto daba lugar a resultados inesperados, generación de objetos duplicados y en algunas ocasiones algún que otro bucle infinito.

Finalmente, el uso de librerías de terceros, como la utilizada para el selector de color (Farbstastic) no ha resultado ser tan simple si al utilizarla no te era suficiente con los valores por defecto. Tras recorrer muchos foros por internet buscando la manera de lograrlo he conseguido redefinir la función 'callback' en la inicialización del componente.

## Capítulo 5

# Pruebas

Las pruebas que he realizado para comprobar que el funcionamiento era correcto han sido a través de la consola de Google devTools. Aplicando puntos de interrupción me permitía depurar el código y obtener los valores deseados en todo momento. Gracias a esta herramienta, la corrección de errores ha sido rápida y precisa.

Por último, a la hora de realizar el envío de emails, la forma de comprobar que el envío era correcto era viendo que el destinatario recibía los mensajes enviados, a parte, también introduje algunas alertas para que saltasen en el caso de que el mensaje no pudiese ser enviado.



## Capítulo 6

# Conclusiones

El interés inicial que tuve a la hora de escoger este proyecto fué el hecho de tener la oportunidad de emplear las capacidades que he ido adquiriendo a lo largo de los diferentes cursos académicos para poder conseguir desarrollar la web con éxito. La programación orientada al desarrollo web siempre ha generado un gran interés en mí, y todo el proceso de creación de un sitio web, desde la parte del servidor (backend) a la parte del cliente (frontend). Esta última (frontend) utiliza tecnologías que apenas conocía por lo que se convirtió en un reto a superar.

El haber realizado este proyecto me ha servido para aprender tanto de manera práctica como teórica.

De manera práctica porque he podido desarrollar una aplicación desde cero, he adquirido muchos conocimientos nuevos y sobre todo, he podido aprender mucho acerca de la tecnología Ajax. Esto podría decirse que es lo que más ilusión me ha hecho, ya que es una tecnología muy útil y usada en la actualidad, y tenía ganas de aprenderla. A parte, mis conocimientos sobre JavaScript y PHP han aumentado notablemente ya que al inicio del proyecto el conocimiento que tenía acerca de estos dos lenguajes era muy básico, pero puedo decir que me he sentido muy cómoda programando con ellos.

De forma teórica porque he aprendido cuáles son las pautas necesarias para poder llevar a cabo un proyecto, a documentarlo como es debido y a saber cómo funciona una empresa.

A pesar de lo mencionado anteriormente, a lo largo del proyecto también me han ido surgiendo dificultades que he tenido que ir solucionando, esto ha sido lo más costoso ya que ha sido un desafío el cual al final he logrado sacar adelante.

En cuanto a lo que al ámbito personal se refiere, me he sentido muy agusto en Aticsoft. Al ser una empresa tan pequeña, el ambiente era muy familiar y esto me ha permitido el poder conocer a cada uno de los miembros del equipo y sentirme como una más.

Por último, los conocimientos y experiencias vividas y adquiridas durante mi paso por Ingeniería Informática no solo me han permitido haber finalizado este proyecto con éxito sino que creo que me han preparado adecuadamente de cara al mundo laboral.



# Anexo

En este apartado, voy a mostrar algunos de los prototipos que realicé a papel antes de empezar con el desarrollo de la aplicación.

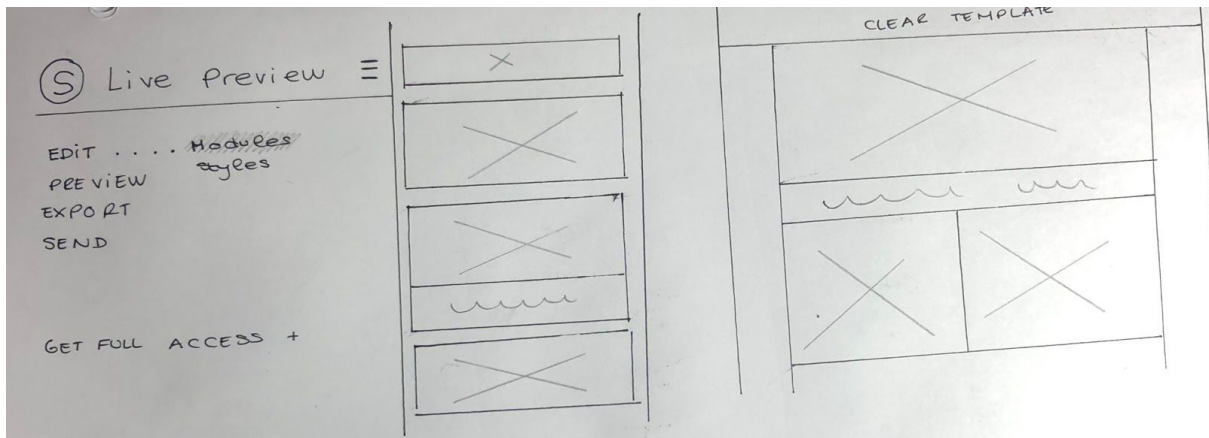


Figura 30: Prototipo de vista de módulos.

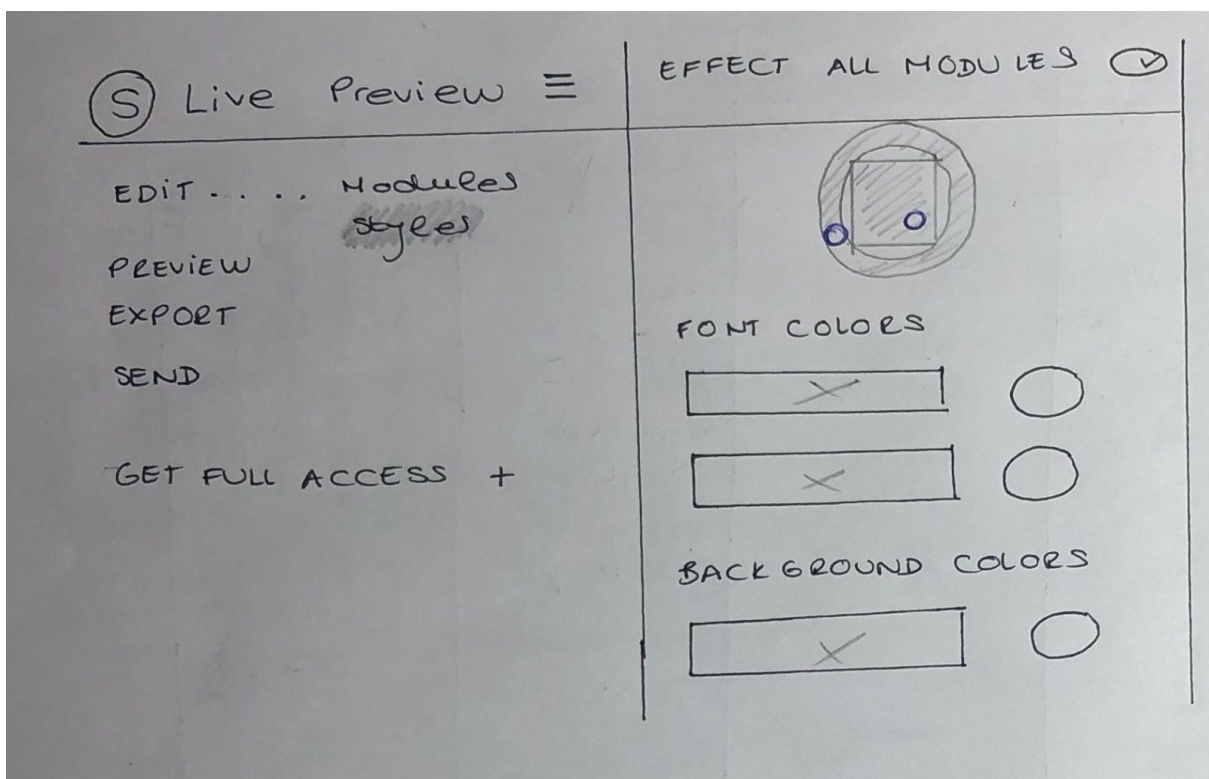
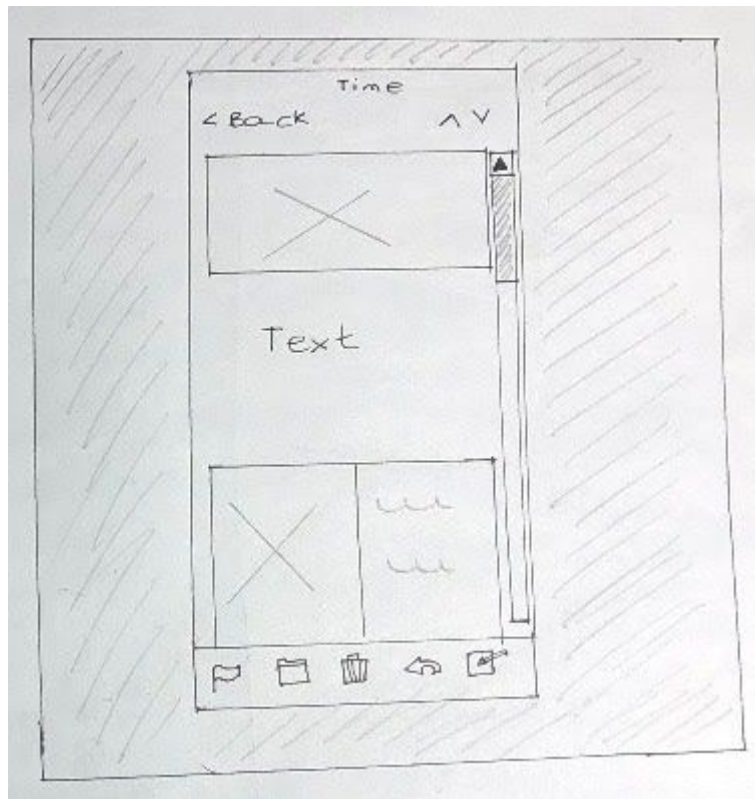
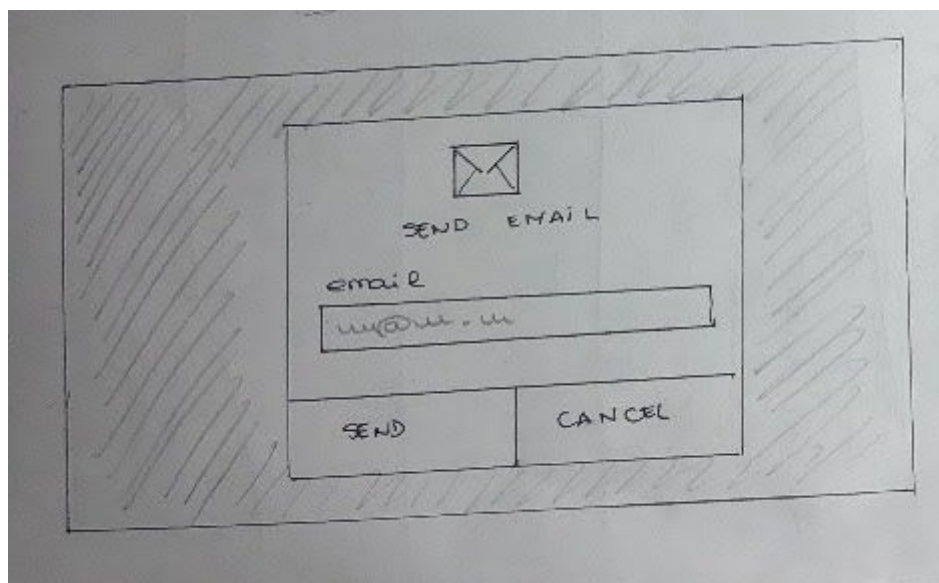


Figura 31: Prototipo de vista de estilos.



**Figura 32:** Prototipo de visualización de pantalla móvil.



**Figura 33:** Prototipo de envío de correo.



# Bibliografía

- [1] Información sobre PHP. <https://www.php.net/manual/es/intro-what-is.php>, último acceso 15/08/2020
- [2] Información sobre SQL. <https://es.wikipedia.org/wiki/SQL> último acceso 15/08/2020
- [3] Ajax. <https://es.wikipedia.org/wiki/AJAX> último acceso 15/08/2020
- [4] Definición HTML. <https://es.wikipedia.org/wiki/HTML> último acceso 15/08/2020
- [5] Definición CSS. [https://es.wikipedia.org/wiki/Hoja\\_de\\_estilos\\_en\\_cascada](https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada) último acceso 15/08/2020
- [6] Bootstrap. [https://es.wikipedia.org/wiki/Bootstrap\\_\(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework)) último acceso 15/08/2020
- [7] Información sobre JQuery. <https://es.wikipedia.org/wiki/JQuery> último acceso 15/08/2020
- [8] Información sobre JQuery-ui. [https://es.wikipedia.org/wiki/JQuery\\_UI](https://es.wikipedia.org/wiki/JQuery_UI) último acceso 15/08/2020
- [9] Información acerca de color-picker Farbstic. <https://acko.net/blog/farbtastic-jquery-color-picker-plugin/> último acceso 15/08/2020
- [10] Información acerca de Eclipse. [https://es.wikipedia.org/wiki/Eclipse\\_\(software\)](https://es.wikipedia.org/wiki/Eclipse_(software)) último acceso 15/08/2020
- [11] Información acerca de Google DevTools. <https://developers.google.com/web/tools/chrome-devtools?hl=es> último acceso 15/08/2020
- [12] Información acerca de Xampp. <https://es.wikipedia.org/wiki/XAMPP> último acceso 15/08/2020
- [13] Información acerca de Mercury. [https://en.wikipedia.org/wiki/Mercury\\_Mail\\_Transport\\_System](https://en.wikipedia.org/wiki/Mercury_Mail_Transport_System) último acceso 15/08/2020
- [14] Información acerca de Filezilla. <https://es.wikipedia.org/wiki/FileZilla> último acceso 15/08/2020
- [15] Información acerca de MySQL-Front. <https://es.wikipedia.org/wiki/HeidiSQL> último acceso 15/08/2020

[16] Herramienta para realizar el diagrama de casos de uso. <https://app.diagrams.net/>  
último acceso 15/08/2020